

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA
EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Trabajo Final de Grado: Memoria

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES
QUE MONITORIZA I CONTROLA EL CONSUMO DE
ENERGÍA DE PROCESOS INDUSTRIALES

Alumno: Adrián Manco Sánchez
Director: Antonio Benito Martínez Velasco
Fecha:
Titulación: Grado en Ingeniería Informática
Centro: Facultad de Informática de Barcelona
Universidad: Universidad Politécnica de Catalunya - BarcelonaTech

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Lista de figuras	1
Introducción	6
Contextualización	6
Empresa	7
Motivación y objetivos.....	10
Motivación.....	11
Objetivos	11
Definición del problema.....	11
Actores implicados	12
Developer	12
Director y jefe de departamento.....	12
Tutor	13
Ponente.....	13
Estado del arte	13
OPC Foundation.....	18
OPC Historical Data Access	19
OPC Data Access.....	19
OPC Unified Architecture.....	20
Alcance del proyecto.....	20
Conceptos generales.....	21
Mejora continua	21
Automatización: definición y tipos.....	24
OEE.....	26
KPI.....	27
ERP	27
MES.....	28
SCADA.....	29
HMI	31
Autómata (PLC)	31
Historian	32
Wonderware.....	33
KEPServer.....	33
OLE for Process Control: qué es y cómo se usa	34
Real Time y Scan Rate en el OPC	39
Organización de la infraestructura: galaxias	41
Industria 4.0.....	42

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Antecedentes	42
Un nuevo paradigma de industria: tecnología que lo sustenta	43
Integración de Tecnologías	50
Cómo utilizan la idea otros ámbitos	52
Comunicaciones: redes para la nueva industria.....	59
Economía y Sociedad.....	81
Gestión Energética Corporativa (Corporate Energy Management).....	87
¿Por qué informatizar la gestión de consumos?	87
EnMPRO: Arquitectura y Escalabilidad.....	88
Entorno de simulación.....	93
KEPServerEX: Simulación con PLC real.....	96
KEPServerEX: Simulador Modbus y Advanced Tags	99
EnMPRO Web: Dashboard de widgets.....	104
Dinámica de Reportes	111
Metodología de trabajo	114
Herramientas de seguimiento	114
Herramientas de validación	114
Posibles obstáculos	114
Planificación general	115
Descripción de tareas	115
Aprendizaje, entrada en contexto y comprensión completa del negocio y la arquitectura	115
Obtención de requisitos y análisis	116
Instalaciones físicas	117
Diseño del sistema de gestión de recursos	117
Entrenamiento del personal y entrega de la solución.....	117
Pruebas y soporte	117
Tiempo estimado.....	118
Planificación original	118
Desarrollo real.....	118
Justificación y explicación de variaciones.....	119
Alternativas y planes de acción asociados	120
Problemas de fechas	120
Dificultades con las herramientas	120
Falta de experiencia	120
Presupuesto del proyecto	120

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Presupuesto original	121
Presupuesto final	121
Justificaciones y diferencias	122
Descripción de las herramientas	122
SIMATIC Energy Manager PRO	122
WonderWare	123
KEPServer	124
Arquitectura de la solución	126
Mercado: Grandes fabricantes de sensores/contadores	127
Electricidad: Socomec	127
Fluidos: Endress & Hauser	127
Contadores y red Modbus	128
Conversión Serie a Ethernet	131
Analyzer (Acquisition), EnMPRO	134
Aplicación: Tarifario	134
APIs	135
Ficheros CSV	136
Serialización	137
Implementación del software	138
Framework y tipo de proyecto	138
Jerarquía de clases	141
Empezando desde abajo: UserControl para Vector	143
Editando la configuración: Form de atributos	145
Escoger qué hacer: Form de menú principal	147
Monitorización: ventana de consumos	150
OPCAutomation: puesta en marcha del cliente OPC	152
EPPlus: creación de la hoja de cálculo	155
Testeo: Comunicación Modbus con Microcontrolador	156
Circuito Digital	156
Circuito Analógico	161
Reflexiones transversales	164
Sostenibilidad y compromiso ambiental	164
¿Has estimado el impacto ambiental que tendrá la realización del proyecto? ¿Te has planteado minimizar el impacto, por ejemplo, reutilizando recursos?	164

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

¿Cómo se resuelven actualmente el problema que quieres abordar (estado del arte)? ¿En qué mejoraría esta solución a las existentes?	164
Sostenibilidad y compromiso económico	165
¿Has estimado el coste de la realización del proyecto?	165
¿Cómo se resuelven actualmente el problema que quieres abordar (estado del arte)? ¿En qué mejoraría esta solución a las existentes?	165
Sostenibilidad y compromiso social.....	165
¿Qué crees que te aportará a nivel personal la realización de este proyecto?	165
¿Cómo se resuelven actualmente el problema que quieres abordar (estado del arte)? ¿En qué mejoraría esta solución a las existentes?	166
¿Existe una necesidad real del proyecto?	166
Utilidad de mis estudios universitarios	166
Bibliografía	168

Lista de figuras

Ilustración 1: Evolución en la Técnica	6
Ilustración 2: Ciclo de Trabajo	8
Ilustración 3: Solución de la Empresa	9
Ilustración 4: Actores implicados	12
Ilustración 5: Arquitectura RPC.....	15
Ilustración 6: Concepto Divide&Conquer	16
Ilustración 7: Collaborative Manufacturing Model Paradigm.....	16
Ilustración 8: Composición en la nube de sistemas ciber-físicos	17
Ilustración 9: Arquitectura IMC-AESOP.....	18
Ilustración 10: Logo de la OPC Foundation	19
Ilustración 11: Gráfica Mejora Continua.....	21
Ilustración 12: Datos Mejora Continua	21
Ilustración 13: Triangulo de Potencias.....	22
Ilustración 14: Ejemplo Automatización.....	24
Ilustración 15: Automatización Discreta	25
Ilustración 16: Automatización Analógica	26
Ilustración 17: ERP	28
Ilustración 18: Pirámide MES	29
Ilustración 19: Niveles de los Sistemas	30
Ilustración 20: SCADA vs MES.....	30
Ilustración 21: Human-Machine Interface	31
Ilustración 22: Programmable Logic Controller	32
Ilustración 23: WonderWare Historian.....	32
Ilustración 24: WonderWare (ArchestrA IDE)	33
Ilustración 25: KEPServerEX	34
Ilustración 26: Funcionalidad OPC	34
Ilustración 27: Componentes de Comunicación	35
Ilustración 28: Visualización de Tags por Dispositivo	35
Ilustración 29: Registros de Direccionamiento	36
Ilustración 30: Quick Client.....	36
Ilustración 31: Cliente OPC en ArchestrA IDE.....	37
Ilustración 32: Mapeo de Atributos.....	37
Ilustración 33: Scripting en Visual Basic	38
Ilustración 34: Object Viewer.....	38
Ilustración 35: Comparación Scan Rates	39
Ilustración 36: Tipos de Sistemas de Tiempo Real	40
Ilustración 37: Jerarquía de la Galaxia	41
Ilustración 38: Revoluciones.....	43
Ilustración 39: Big Data and Analytics.....	44
Ilustración 40: Coworking	44
Ilustración 41: Simulación Virtual de Planta.....	45
Ilustración 42: Industrial Internet of Things.....	46
Ilustración 43: Cybersecurity en Industria	46
Ilustración 44: Cloud en Industria	47

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Ilustración 45: Manufactura Aditiva	47
Ilustración 46: Propiedades de Voxel	48
Ilustración 47: Realidad Aumentada	49
Ilustración 48: Analogía Periféricos-ODBC	49
Ilustración 49: Funcionamiento Web Services	50
Ilustración 50: Panorama de un Sistema Industrial	51
Ilustración 51: Controles de Vuelo	52
Ilustración 52: Fly-by-Wire	53
Ilustración 53: Beneficios ABS	54
Ilustración 54: Mecanismo ABS	54
Ilustración 55: Zona Libre de Personas	55
Ilustración 56: Transporte Autónomo	55
Ilustración 57: Gruas Autónomas	55
Ilustración 58: Semi-Automated Mason	56
Ilustración 59: Built Robotics	56
Ilustración 60: Bucle de Detección	57
Ilustración 61: Visualización de la Red	57
Ilustración 62: Plataformas Móviles	58
Ilustración 63: Estación Robótica Autónoma	58
Ilustración 64: Ejemplo de Red Industrial	59
Ilustración 65: Justificación de las Redes Industriales	60
Ilustración 66: Principios de las Redes	60
Ilustración 67: Sincronización Temporal	62
Ilustración 68: Franjas de Prioridades	62
Ilustración 69: Solapamiento de Prioridades	63
Ilustración 70: Guard Band	63
Ilustración 71: Preempción	64
Ilustración 72: HSR	65
Ilustración 73: PRP	65
Ilustración 74: Categorías de Diseños	66
Ilustración 75: Objetivos y Propuestas	67
Ilustración 76: Barreras y Requerimientos	67
Ilustración 77: Jerarquización en Zonas	68
Ilustración 78: Niveles por Zona	69
Ilustración 79: Zonas y Cells	69
Ilustración 80: Flujo de Control	70
Ilustración 81: DMZ y Elementos por Nivel	71
Ilustración 82: Diseño DMZ	72
Ilustración 83: Separación Core-Distribution	73
Ilustración 84: Core-Distribution Colapsados	74
Ilustración 85: Disposición Lineal	75
Ilustración 86: Disposición de Estrella Redundante	75
Ilustración 87: Esquema General	76
Ilustración 88: Evolución Ciberataques	77
Ilustración 89: Riesgos IoT	78
Ilustración 90: Riesgos Blockchain	79
Ilustración 91: Vulnerabilidades Inteligencia Artificial	80
Ilustración 92: Edge Computing	81

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Ilustración 93: Susceptibilidad a Automatizarse	82
Ilustración 94: Sensibilidad a Evolucionar	82
Ilustración 95: Previsión de Pérdida de Empleo Humano.....	83
Ilustración 96: 5 Puntos de Reflexión.....	84
Ilustración 97: Equipo Industrial	84
Ilustración 98: Manifestaciones en Rotterdam	86
Ilustración 99: Cadena de Consecuencias.....	86
Ilustración 100: Efectos CEM	88
Ilustración 101: Pirámide EnMPRO	89
Ilustración 102: Arquitectura EnMPRO	90
Ilustración 103: Jerarquía EnMPRO	91
Ilustración 104: Niveles del Sistema CEM	92
Ilustración 105: Diferentes Periferias.....	93
Ilustración 106: Entorno General de Simulación	94
Ilustración 107: Arquitectura de la Simulación	95
Ilustración 108: Autómata	95
Ilustración 109: Conexión del Autómata a la Red	96
Ilustración 110: Conectividad con el PLC.....	97
Ilustración 111: Programación de Contador	97
Ilustración 112: Programación de Borrado.....	98
Ilustración 113: Configuración PLC (I)	99
Ilustración 114: Configuración PLC (II)	99
Ilustración 115: Funciones de Simulación	100
Ilustración 116: Justificación Advanced Tags	101
Ilustración 117: Evolución de la Señal (I)	101
Ilustración 118: Evolución de la Señal (II)	101
Ilustración 119: Advanced Tag	102
Ilustración 120: Configuración de Advanced Tag (Link)	102
Ilustración 121: Comprobación del Flujo de Datos (I)	103
Ilustración 122: Comprobación del Flujo de Datos (II)	103
Ilustración 123: Comprobación Final (I)	103
Ilustración 124: Comprobación Final (II)	103
Ilustración 125: Tags creados.....	104
Ilustración 126: Tags en el Quick Client.....	104
Ilustración 127: Widget Dashboard	105
Ilustración 128: Acquisition para la Simulación.....	105
Ilustración 129: Gauge (medidor).....	106
Ilustración 130: Semáforo	106
Ilustración 131: Configuración medidor/semáforo	106
Ilustración 132: Chart	107
Ilustración 133: Pie Chart	108
Ilustración 134: Table.....	108
Ilustración 135: Valores	109
Ilustración 136: Texto	109
Ilustración 137: Imagen	109
Ilustración 138: Heatmap	110
Ilustración 139: Diagrama Sankey	110
Ilustración 140: Relaciones Sankey	111

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Ilustración 141: Nodos Sankey	111
Ilustración 142: Base de Datos de Producción	112
Ilustración 143: Conjunto de Datos	112
Ilustración 144: Reporte	113
Ilustración 145: Planificación Temporal Original	118
Ilustración 146: Gantt Original	118
Ilustración 147: Dependencias Originales	118
Ilustración 148: Planificación Temporal Definitiva	119
Ilustración 149: Gantt Definitivo	119
Ilustración 150: Dependencias Definitivas	119
Ilustración 151: Presupuesto Hardware Original	121
Ilustración 152: Presupuesto Software Original	121
Ilustración 153: Presupuesto RRHH Original.....	121
Ilustración 154: Presupuesto Hardware Final	121
Ilustración 155: Presupuesto Software Final	122
Ilustración 156: Presupuestos RRHH Final.....	122
Ilustración 157: Logo SIEMENS.....	123
Ilustración 158: Logo WonderWare.....	123
Ilustración 159: Logo Kepware-KEPServer.....	124
Ilustración 160: Concepto KEPServerEX	125
Ilustración 161: Tags Avanzados	125
Ilustración 162: Alarmas y Eventos	126
Ilustración 163: Arquitectura de la Solución.....	126
Ilustración 164: Socomec	127
Ilustración 165: Endress & Hauser	127
Ilustración 166: SENTRON 7KM PAC3200	128
Ilustración 167: Modbus.....	128
Ilustración 168: Bloques de Modelo de Datos	129
Ilustración 169: Representación Direcciones Modbus.....	129
Ilustración 170: Prefijos Modbus.....	129
Ilustración 171: Topología Daisy Chain	131
Ilustración 172: Conversor Serial a Ethernet.....	131
Ilustración 173: Ejemplo de Conversión Serial-Ethernet.....	132
Ilustración 174: Conversor Industrial RS-232 a Ethernet.....	132
Ilustración 175: Conversor Inalámbrico RS-232 a Ethernet	133
Ilustración 176: Conversor PCB RS-232 a Ethernet	133
Ilustración 177: Acquisition y EnMPRO	134
Ilustración 178: Concepto de API	135
Ilustración 179: Comparación Síncrono-Asíncrono	136
Ilustración 180: Correspondencia Excel-CSV	136
Ilustración 181: Formatos de Serialización	137
Ilustración 182: Ejemplo de Solución en VS	139
Ilustración 183: Concepto UserControl	140
Ilustración 184: UserControl en la Aplicación.....	140
Ilustración 185: Flujo de funcionamiento	141
Ilustración 186: Clases de datos.....	142
Ilustración 187: Diagrama de Clases.....	143
Ilustración 188: Flujo de datos desde UserControl.....	143

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Ilustración 189: UserControl para Vector	144
Ilustración 190: Guardado de Tramos	145
Ilustración 191: Edición de configuraciones.....	146
Ilustración 192: Creación del nuevo Vector	146
Ilustración 193: Guardado de vectores a CSV	147
Ilustración 194: Form de elección de modo	148
Ilustración 195: Uso de Tag en el ListView	149
Ilustración 196: Vinculación mediante Tags.....	149
Ilustración 197: Añadido de filas y vinculación de Tags.....	150
Ilustración 198: Pantalla de monitorización.....	151
Ilustración 199: Declaración de las estructuras necesarias	151
Ilustración 200: Añadido de puntos/lecturas	152
Ilustración 201: Conexión al servidor OPC	152
Ilustración 202: Inicialización del grupo OPC	153
Ilustración 203: Añadido de un ítem OPC	153
Ilustración 204: Método y evento de lectura asíncrona	154
Ilustración 205: Monitorización activada y evento síncrono	154
Ilustración 206: Borrado de grupos y desconexión de servidores.....	155
Ilustración 207: Generación de la hoja de cálculo	155
Ilustración 208: Hoja de cálculo generada.....	156
Ilustración 209: Arquitectura de Testeo.....	157
Ilustración 210: Circuito Analógico	157
Ilustración 211: Modbus Digital.....	158
Ilustración 212: Comprobación de Coils en ModbusMAT	159
Ilustración 213: Mapeo Digital	159
Ilustración 214: Monitorización del Circuito Digital.....	160
Ilustración 215: Pulse Width Modulation.....	161
Ilustración 216: Modbus Analógico	162
Ilustración 217: Luminosidad del LED	162
Ilustración 218: Mapeo Analógico	163
Ilustración 219: Monitorización del Circuito Analógico	163

Introducción

En este apartado se explicará el por qué y el cómo de este proyecto, de manera que la lectura de la memoria sea lo más provechosa y comprensiva posible.

Contextualización

Es importante tener conocimiento del contexto en el que se encuentra el tema del trabajo, así como el estudiante que lo lleva a cabo.

Desde que existe cualquier cosa considerable como manufactura, se ha buscado maximizar la eficiencia del proceso productivo, ya que como todo lo vinculado a la actividad comercial, lo que se quiere es maximizar los beneficios, reduciendo los costes derivados de la actividad productiva lo máximo posible.

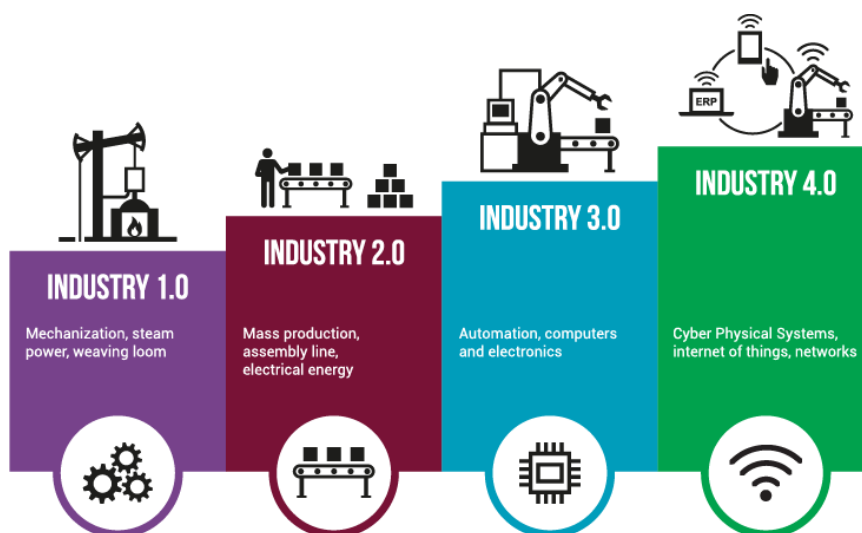


Ilustración 1: Evolución en la Técnica

A finales del siglo XVII, la introducción de maquinaria propulsada a vapor o agua comportó una verdadera revolución en cuanto al proceso productivo y la industria en general, pero no se limitó a acelerar la fabricación. Aunque la crecida de la productividad fue dramática, permitiendo a las fábricas producir más unidades con menos esfuerzo humano, entraron en juego factores antes ignorados, derivados del uso de la nueva maquinaria. Ahora se tenía que tener en cuenta el proceso de transporte del agua, el mantenimiento del equipo y la formación del personal, solo por nombrar algunos de los factores introducidos.

Posteriormente, la introducción de la electricidad como fuente de energía comportó la aparición no solo de los factores derivados del consumo del propio recurso, sino de aquellos directamente relacionados con el uso de la nueva maquinaria, disponible gracias a los avances en la técnica. Los computadores, en concreto, dieron una nueva perspectiva sobre cómo se podía monitorizar la producción, y también sobre cómo se podía controlar mediante las propias máquinas.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Siempre ha sido necesario controlar y monitorizar cualquier proceso productivo, pero ya no se trataba solo de supervisar que los trabajadores produjesen a mano un número concreto de unidades de producto, y con una cierta calidad. La cuestión ahora era que las nuevas máquinas habían cambiado por completo el paradigma de lo que se conocía como control y monitorización industrial, y los computadores tendrían que jugar un rol clave para intentar solucionar esta necesidad. Eran los únicos capaces de procesar el flujo de datos generado en un tiempo conveniente, y de tomar decisiones en tiempo real, y de manera simultánea en diversas máquinas.

Fue entonces cuando surgió la idea del MES (Manufacturing Execution System). El concepto es sencillo: igual que cuando utilizamos un ordenador tenemos un Sistema Operativo, encargado de abstraer al máximo posible al usuario de tener que interactuar directamente con el hardware, los sistemas de automatización como el MES permiten controlar y monitorizar los recursos de la industria sin tener que realizar acciones como ir a las máquinas a tomar medidas, o pulsar físicamente un botón para cerrar o abrir una válvula, entre otras.

Es este, entonces, un sistema crítico para mantener la competitividad. Con la crecida de la llamada Industria 4.0, cada vez más compañías han adoptado algún tipo de sistema de automatización industrial, ya que es imperativo si no se quiere perder mercado. De manera sencilla: la eficiencia que puede llegar a conseguir uno de estos sistemas permite que una empresa produzca más unidades de un producto, puede que invirtiendo menos tiempo y dinero, y lo venda a un precio igual o inferior a productos equivalentes en el mercado, todo esto manteniendo un beneficio razonable. Es entonces evidente que aquella industria que no evolucione en este sentido y tenga que competir con otras que sí que lo han hecho, está destinada a reubicarse en el mercado o a modernizarse, si no quiere quedar apartada.

Empresa

En tanto que un proyecto en modalidad B, se realiza dentro de un CCE en una empresa, de la cual se da una descripción.

Este proyecto se realiza dentro de un Convenio de Colaboración Educativa (CCE) con una empresa dedicada al desarrollo e integración de soluciones tecnológicas para la industria, especializada en desplegar sistemas que cubren las disciplinas, principalmente, de la Informática, la Automatización y el Mantenimiento Industrial, siendo capaces de crear valor añadido para el cliente. Consta de una sede central en Montcada i Reixac (Barcelona), pero cuenta con delegaciones en Malgrat de Mar i Bucarest (Rumania). Entre sus clientes constan algunas de las firmas más conocidas en diversos ámbitos, sobretudo el alimentario: Cobega, SEAT, Cerveza Damm, Mitsubishi Electric, Schneider Electric, Cafés Marcilla, etc. Son solo algunos ejemplos, y cuenta con sistemas desplegados tanto en España como en el extranjero.

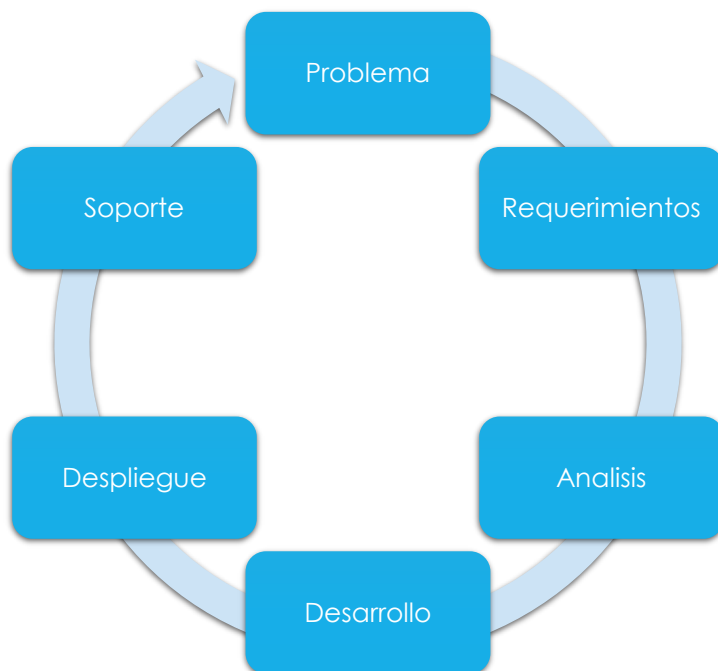


Ilustración 2: Ciclo de Trabajo

La dinámica de trabajo de la empresa se rige por un ciclo general de trabajo. Un cliente les contacta con un problema, que usualmente no saben cómo resolver. Una vez se define el problema, se recogen los requerimientos específicos a la hora de solucionarlo. Con toda la información recogida, los equipos de desarrollo se reúnen para analizar cómo abordar el proyecto. Una vez se acuerda una estrategia, llega el momento de desarrollar, es decir, de implementar la solución que satisfará las necesidades del cliente. Después de testear y cerciorarse de que el fruto del desarrollo es adecuado, se pasa al despliegue y la integración de la solución con las tecnologías ya existentes en las instalaciones del cliente. Por último, una vez se cierra el proyecto, se acuerda un tiempo de soporte técnico en el cual el cliente puede contactar con el personal de la empresa para solucionar los problemas y dudas que puedan surgir.

En sus 15 años de experiencia en el sector, se ha encargado de hacer llegar a los clientes una propuesta de automatización que permita hacer crecer la eficiencia de sus procesos productivos, así como historizar toda una serie de datos recogidos mediante los sistemas desplegados en planta, para su posterior análisis y uso en el proceso de Mejora Continua.

El servicio principal que ofrece a potenciales clientes interesados a dar el salto a la industria 4.0 es el siguiente (se ofrece una explicación más detallada de algunos de los conceptos en la sección "Conceptos generales" de la Introducción):

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

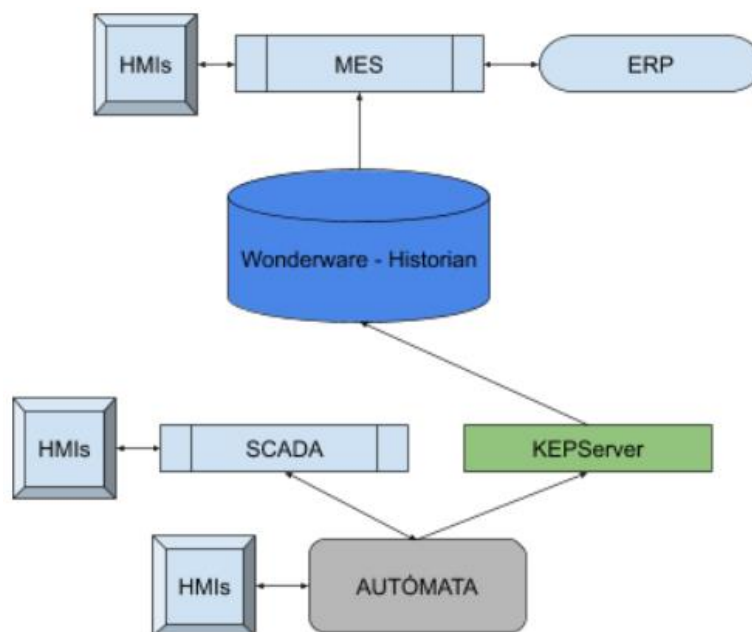


Ilustración 3: Solución de la Empresa

En el plano más bajo, estarían los autómatas, es decir, las computadoras encargadas de controlar las máquinas de producción, y de donde los automatistas extraen las señales que el sistema necesita. A ellas van conectados tanto los SCADA, en comunicación bidireccional, como el KEPServer, en comunicación ascendente con Historian. Éste último, a su vez, se encarga de conectarse al MES, para que se pueda proceder al tratamiento de los datos. De manera simultánea a toda esta arquitectura, puede existir toda una infraestructura de HMIs conectados a varios puntos del sistema, y un ERP que conecte al personal comercial con los datos recogidos en planta.

Además de esta propuesta de automatización, la organización está abierta a toda una serie de proyectos, de mayor o menor envergadura, siempre relacionados con la industria. A continuación, se listan algunos ejemplos:

- Desarrollo de sistemas de inventario para agilizar la gestión de salidas y entradas de almacén, y facilitar al personal el mantenimiento de un registro actualizado y conveniente de las existencias de la empresa.
- Despliegue de soluciones de final de línea: creación automatizada de etiquetas vinculadas a la base de datos de producción, automatización del proceso de pegado de etiquetas de pallet, etc.
- Integración de puntos concretos del proceso productivo a la lógica del MES, así como soluciones de automatización para puntos concretos de la línea de producción.
- Integración de herramientas de gestión dentro de la automatización de los procesos productivos.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

En el ámbito educativo y de formación, la empresa colabora con diversas organizaciones para acoger estudiantes en prácticas, y ofrecer cursos de especialización, en su ámbito de trabajo. Para ello desarrollaron una plataforma que permite realizar estos estudios de manera remota, aunque disponen de aulas para formación presencial. Algunos de los cursos ofertados son los siguientes:

- Automatización Industrial: Visión Artificial, Profinet, Profibus, Hidráulica y Electroneumática Industrial, etc.
- Tecnologías de la Información: Ethernet Industrial y Switches Gestionables, Virtualización, etc.
- Informática Industrial: Arduino Industrial (niveles básico y avanzado), Lean Manufacturing Integrado en MES, etc.
- Mantenimiento Industrial: Análisis de Vibraciones, Operador Industrial de Calderas, Electricidad Industrial (niveles básico y avanzado).

Motivación y objetivos

Un TFG supone horas de dedicación, más aún si supone prácticas en una empresa. ¿Por qué realizar este proyecto?

En la universidad se adquiere toda una serie de conocimientos tanto teóricos como prácticos, que conforman la base de mi educación como ingeniero. Aunque orientados a ejemplificar situaciones reales, los métodos evaluativos encontrados a lo largo de los años de grado no dejan de ser eventos académicos, que no reflejan las situaciones reales encontradas en la vida laboral. Teniendo en cuenta esto, y las facilidades que ofrece la FIB para realizar conjuntamente prácticas remuneradas en empresa y el TFG, me pareció una oportunidad interesante de ahorrar dinero de matrícula, poner el broche final a mi carrera, y de paso adquirir experiencia laboral valiosa de cara a mi futuro.

En cuanto al tema de mi proyecto, viene a raíz de contactar con mi ponente. Como profesor de Robótica, le contacté porque veía en su campo una oportunidad para profundizar mis conocimientos, pero mi itinerario de asignaturas hacía difícil mi integración en los equipos con los que él participaba. Posteriormente me introdujo el concepto de Informática Industrial, y después de investigar por cuenta propia, decidí que era un ámbito al cual no me importaría dedicar algunos meses de estudio.

Una vez empecé a estar asentado en la empresa, la temática de mi TFG se me empezó a aparecer clara. Actualmente existe una preocupación muy grande en cuanto al medio ambiente, entonces, ¿por qué no encauzar dentro de la informática industrial un proyecto que tenga esto en cuenta? De esta manera, adquiriría conocimiento técnico en el ámbito, y dedicaría el producto de mi investigación a una finalidad de transcendencia actual y futura.

También quería utilizar este periodo de prácticas para decidir si quería realizar o no un máster, y en caso de querer, cuál. A día de hoy, puedo afirmar que me

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

ha servido para decidir que sí que quiero seguir con mis estudios y especializarme, probablemente en la rama de redes y comunicaciones, aunque no sé si llevaré esta formación al ámbito de la informática industrial.

Motivación

- Finalizar mis estudios en el Grado en Ingeniería Informática.
- Adquirir experiencia laboral.
- Conocer un ámbito concreto de la informática.
- Desarrollar un proyecto de relevancia.
- Hacer crecer mi CV.
- Ayudarme a decidir mi futuro.

Objetivos

- Leer consumos de manera fiable y temporalmente conveniente
- Crear simulaciones que faciliten la implantación de estos sistemas
- Permitir informatizar tarifas energéticas
- Dar idea a los usuarios de cómo de grande es su huella ambiental
- Diseñar una arquitectura aplicable a cualquier tipo de industria

Definición del problema

La finalidad de un TFG es desarrollar una solución o una investigación enfocada en un problema concreto. El que se trata en éste es muy determinado.

Como todo sistema complejo, el que nos ocupa está formado por todo un conjunto de partes, que, por medio de una gestión jerarquizada, tratan un problema mayor: gestión de línea, gestión de producción, control financiero, etc... Este proyecto se encargará de desarrollar el sistema de gestión de energía de una empresa, conocido por sus siglas en inglés como CEM (Corporate Energy Management). Este subsistema es el encargado de trasladar la información de los consumos energéticos en tiempo real al sistema de gestión general, hacer uso de contadores para leer los datos de consumo, y generar los eventos y alarmas necesarios, según las necesidades concretas de cada despliegue. También se generan reportes y gráficas para ilustrar de manera intuitiva la evolución del uso de recursos. El software SIMATIC Energy Manager PRO de Siemens se encargará de integrar los datos recogidos, y de realizar las tareas de cálculo y visualización convenientes, mientras que una aplicación, desarrollada

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

especialmente para el proyecto, se encargará de exportar datos para EnMPRO, aparte de implementar algunas funcionalidades de éste, de manera simplificada. Esto hará el sistema accesible tanto a personal altamente cualificado, como a cualquier operario de planta. Cabe decir que aquí se menciona EnMPRO por requisitos del proyecto, pero cualquier herramienta equivalente, que acepte el mismo tipo de importación de datos que éste, puede beneficiarse de los resultados de este proyecto.

Gracias a los CEM podemos, entonces, automatizar todos los aspectos referentes al consumo energético, y poner la monitorización y el control de los recursos a la altura de las del resto de procesos industriales.

Actores implicados

Hace falta definir que entidades se ven afectadas por el desarrollo de este proyecto, es decir qué papel juegan y qué provecho sacan.



Ilustración 4: Actores implicados

Developer

Estudiante en prácticas que realiza el proyecto de acuerdo con los requerimientos. Se encarga de formarse, documentarse, implementar y testear el progreso del proyecto, además de redactar la memoria final. Cumple con los deadlines tanto del ponente como del director del proyecto, y combina la tarea de aprendizaje en la empresa con el desempeño laboral y la implementación del proyecto.

Su interés principal es la adquisición del título de graduado, completar su formación y realizar un proyecto trascendente y de utilidad actual y futura.

Director y jefe de departamento

Representantes de la empresa, que proporcionan los requisitos al estudiante para la realización de sus proyectos. Encajan la tarea del estudiante dentro de su actividad comercial, esperando que les reporte algún beneficio más allá de la estancia de éste, ya sea a nivel económico o de conocimiento. Se encargan de asignar un tutor al estudiante, que le ayude a asentarse y a formarse, para realizar su proyecto de la mejor manera posible. Establecen deadlines y ofrecen soporte durante la realización del TFG.

Tutor

Miembro de la plantilla de la empresa encargado de guiar al estudiante durante todas las fases de aprendizaje del proyecto. Es la primera persona a la que tiene que acudir, ya que se le supone conocimiento tanto técnico como del negocio al cual el proyecto es aplicable.

Ponente

Miembro del claustro de la UPC que tutoriza, de cara a la universidad, la realización del proyecto. Colabora con el alumno para realizar todos los procedimientos formales evaluativos, establece objetivos a cumplir de manera progresiva y convoca reuniones de seguimiento.

Estado del arte

A la hora de profundizar en un tema, hay que tener en mente hasta dónde se ha desarrollado éste en el momento de la investigación.

En el mercado hay multitud de alternativas a la hora de escoger una solución de automatización industrial. Para desplegar un sistema eficaz, hace falta una serie de dispositivos, tanto hardware como software, encargados de crear la infraestructura necesaria. La heterogeneidad de estos elementos hace que no haya una arquitectura estandarizada a la hora de desplegar este tipo de soluciones, pero sí que hay guías generales que se suelen seguir, y grandes marcas de las que se puede tomar ejemplo.

En cuanto a la interacción máquinas/computadores, KEPServer es el software más utilizado. Entre las funcionalidades que lo hacen el más popular, destacan su gran cantidad de drivers, la calidad de su implementación interna, la facilidad de crear simuladores, o la cantidad de opciones en cuanto a alarmas, historización o IoT.

Tenemos, entonces, que los sistemas de automatización se pueden (y de hecho se suelen) construir haciendo que varias partes, de fabricantes potencialmente diferentes cooperen. Pese a esto, hay marcas que ofrecen soluciones integrales capaces de soportar toda la lógica del sistema. Es el caso de WonderWare o Microsoft, que ofrecen suites MES completas.

La empresa en la que se ubica este proyecto decidió elaborar una solución propia a partir de la integración de software de terceros como partes de KEPServer o WonderWare con aplicativos propios. Mediante el uso mixto de tecnologías, abaratan el coste que supone para un hipotético cliente adquirir una solución completa de una de las dos marcas previamente mencionadas. La lógica de más bajo nivel es confiada a los softwares propietarios, y las tareas de control de las máquinas (SCADA) y visualización de datos es implementada por equipos e de desarrollo propio.

El estado del arte de la automatización tiende a sistemas que confían en la arquitectura orientada a servicios. En cuanto a este último concepto, el proyecto IMC-AESOP es la frontera de investigación en este ámbito: se encarga de tratar los principales problemas encontrados en los sistemas de

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

automatización a gran escala hoy en día, e intenta aplicar la arquitectura orientada a servicios para solucionarlos.

Sumarizando, los problemas de los sistemas actuales son los siguientes:

- Pueden tener decenas de miles de conexiones en una determinada planta, lo que hace puede hacer pesados los procesos de control y el tratamiento de los datos.
- Tienen que soportar numerosas peticiones de datos de manera simultánea, arriesgándose a causar flooding (inundación) de la red y a saturar los equipos de trabajo.
- Constan de equipos con rendimientos, funcionalidades, requerimientos y consumos heterogéneos. Las tareas de manejo y soporte resultan inviables si los operarios tienen que ser expertos en cada una de las marcas que participan del sistema.
- Sufren las consecuencias de utilizar simultáneamente equipos punteros y maquinaria prácticamente desfasada. Las tareas de integración y actualización se dificultan, así como la interacción entre los dispositivos, según la brecha temporal aumenta.

Para ello, el modelo IMC-AESOP pretende:

- Permitir a los dispositivos mostrar sus funcionalidades como servicios.
 - De esta manera, el acceso a sus capacidades se simplifica, y se puede aplicar una metodología parecida a los Remote Procedure Call. La idea es que un dispositivo publique sus funcionalidades, es decir, qué ofrece, cómo lo ofrece, y cómo tiene que ser solicitado. Así, una máquina cliente puede enviar una solicitud de la manera adecuada a través de la red a una máquina server, y ésta, después de ejecutar el servicio, retorna los resultados.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

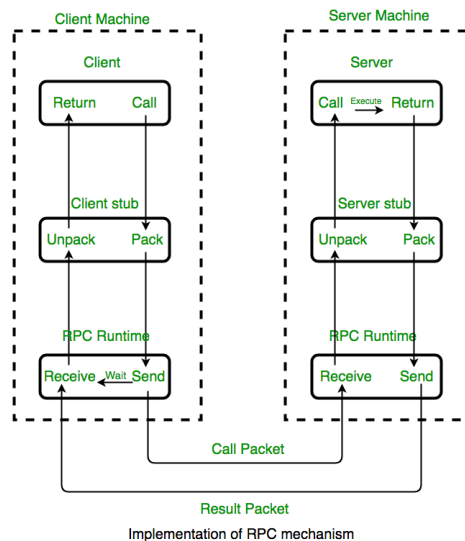


Ilustración 5: Arquitectura RPC

- Permitir a cualquier equipo capaz de invocar servicios tomar parte en el proceso de control i monitorización.
 - Al descentralizar el acceso al sistema, cualquier punto de la planta debería de ser accesible. Se eliminan los puntos ciegos y se facilita la transmisión en tiempo real de los datos. Esta idea está relacionada con el Edge Computing, del cual se hablará más adelante.
- Hacer posible la agregación o composición de servicios con el propósito de generar nuevos subsistemas de control (SCADA).
 - Distinguir varias partes discretas de un gran sistema facilita la operación de este, mediante la atención aplicada a cada una de sus partes. Intentar controlar un sistema al completo puede resultar apabullante, pero desarrollar una metodología para gestionar uno o unos cuantos pequeños sistemas a la vez suele resultar más efectivo. Es una estrategia históricamente aplicada a una gran cantidad de ámbitos, desde la computación (Divide&Conquer) a tácticas militares o políticas.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

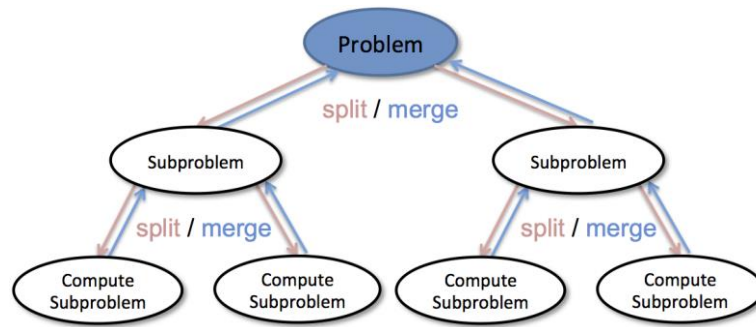


Ilustración 6: Concepto Divide&Conquer

- Conectar mediante servicios la planta comercial con la de producción.
 - Permitir la interacción del sistema ERP de la compañía con los SCADA y MES que se usan en planta hace que la automatización y la visibilidad de la compañía sea total. Mediante esta comunicación, los sistemas de producción pueden invocar, por ejemplo, un servicio que cargue órdenes de abastecimiento en el ERP, o de manera inversa, el ERP puede crear órdenes de producción una vez se cierra una venta.

Aunque centrado en resolver los problemas ampliamente encontrados en la industria, IMC-AESOP va más allá, y conceptualiza algunas de las ventajas de la implantación de sus directrices, creándose los axiomas sobre los que se fundamentará la industria del futuro.

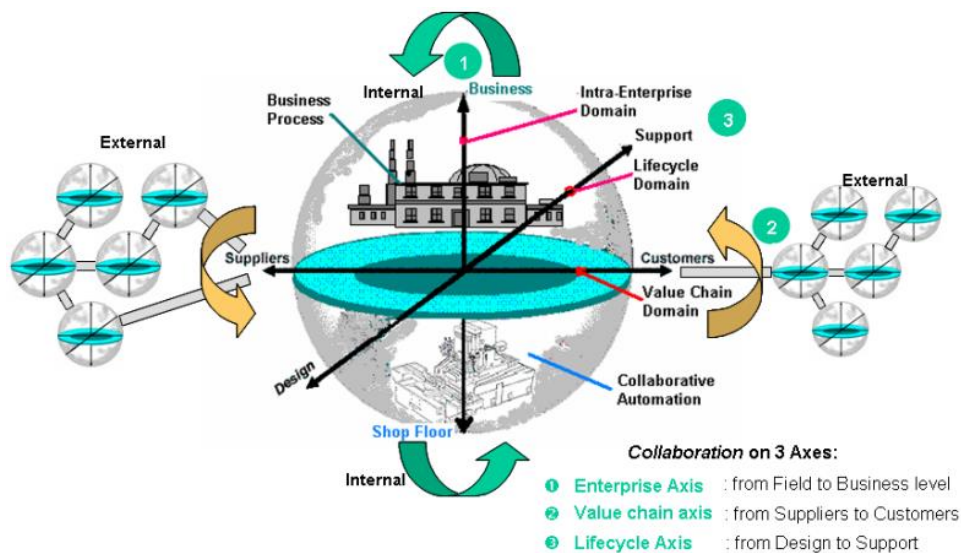


Ilustración 7: Collaborative Manufacturing Model Paradigm

Un ejemplo es el Paradigma del Modelo de Manufactura Colaborativa (CollaborativeManufacturing Model Paradigm). Partiendo de la idea de un Sistema de Sistemas, divide en tres ejes la colaboración industrial:

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

- Eje Corporativo: del nivel de Planta al nivel Empresarial.
- Eje de Cadena de Valor: de Proveedores a Clientes.
- Eje de Ciclo de Vida: de Diseño a Soporte.

Esta visión tridimensional del modelo permite apreciar la naturaleza cíclica no solo de los procesos de desarrollo, sino también de la interacción entre las diferentes partes de la actividad industrial: proveedores que suministran a sus clientes, que a su vez pueden ser proveedores de otro, plantas que se alimentan del nivel comercial y también les envían información, etc.

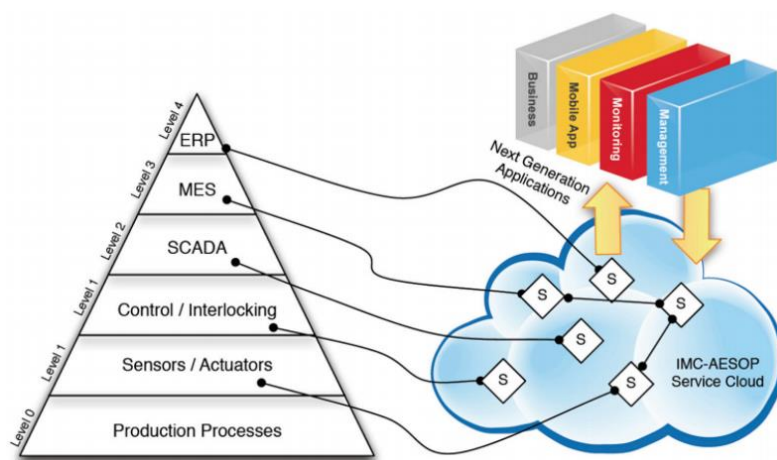


Ilustración 8: Composición en la nube de sistemas ciber-físicos

Se plantea no solo publicar como servicios las capacidades de cada capa del sistema, sino alojar dichos servicios en un cloud, que liberaría a las compañías de tener que adquirir y mantener hardware propio, a la vez que les ofrecería las altas prestaciones y facilidades de los servicios en la nube. Como efecto colateral, la adaptación a gran escala de esta arquitectura podría abrir un nicho de negocio a los proveedores de cloud.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

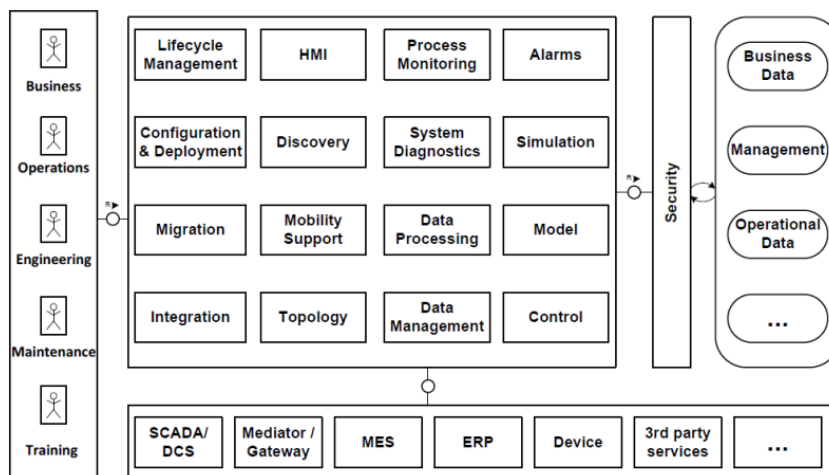


Ilustración 9: Arquitectura IMC-AESOP

La ilustración muestra la arquitectura IMC-AESOP. A la izquierda tenemos los agentes que pueden querer acceder a los datos. En la sección de la derecha se muestran los datos que pueden ser accedidos, así como las comprobaciones de seguridad que les preceden. Para distinguir los diferentes tipos de servicio, se distinguen diferentes grupos, de manera que se puedan aplicar políticas de manera adecuada a las necesidades de cada uno. Por último, en la sección inferior, encontramos todos los dispositivos y aplicaciones que pueden invocar los servicios, desde los SCADA, a otros servicios de terceros.

Volviendo al objetivo de este TFG, hay que aclarar que en tanto que un estudio de una de las muchas partes que conforma un sistema de automatización industrial, no busca innovar en ningún aspecto ni hacer avanzar la frontera de conocimiento en el ámbito. En vez de eso, pretende unir una tendencia técnica creciente con una preocupación en aumento por el medio ambiente, lo que de cara a las empresas es muy beneficioso: la automatización y la bajada de gasto energético aumenta los beneficios, mientras que reducir el impacto ambiental mejora la imagen de la compañía.

OPC Foundation



Ilustración 10: Logo de la OPC Foundation

Es un consorcio industrial encargado de definir y mantener estándares para la automatización industrial y la conectividad abierta entre dispositivos. Éstos estándares especifican la comunicación de datos industriales, así como alarmas y eventos, entre otros. En 1996, el grupo de empresas que formaban el equipo operativo de OPC Foundation publicó el primer estándar para OPC. A día de hoy, se sigue trabajando en la actualización y mejora de los estándares, y se colabora con otras organizaciones con propósitos similares.

Algunos de los estándares más notables son los siguientes:

OPC Historical Data Access

Define cómo se tiene que realizar la comunicación entre dispositivos que proveen de datos históricos, ya sean en crudo, interpolados o agregados, es decir, resultados de cálculos. En esencia, HDA sigue la misma filosofía que OPC Data Access, pero con la diferencia que mientras que HDA está pensado para manejar datos almacenados, DA tiene el propósito de manejar datos en tiempo real.

Como DA, utiliza la tecnología DCOM de Microsoft, por las funcionalidades de seguridad y autenticación que incluye, así como la facilidad de integración, con equipos Windows que ofrece.

OPC Data Access

Especifica cómo comunicar datos en tiempo real desde dispositivos de adquisición tipo PLC, a interfaces como los HMI, los SCADA o los ERP/MES. El alcance de DA se limita a datos en tiempo real y eventos (OPC HDA se encarga de los datos históricos), y define tres atributos que toda señal transmitida tiene que tener: el valor, la calidad, y el timestamp.

Los tres atributos mencionados son obligatorios, y si el servidor observa que el cliente no es capaz de generar, por ejemplo, el timestamp, tiene que generarlo él mismo. Esta atención por el comportamiento de los otros dispositivos en la comunicación hace que los datos se envíen y reciben adecuadamente.

OPC Unified Architecture

Surgido como evolución del modelo OPC clásico, OPC UA surgió con el propósito de solucionar las limitaciones de alcance que el paradigma anterior tenía. UA no solo se encarga de llevar datos de sensores a aplicaciones tipo SCADA o MES, sino que hace que la información fluya entre todas las capas de la empresa, desde los sistemas de gestión financiero, a ERPs y hojas de Excel.

Según la OPC Foundation, los objetivos son los siguientes:

- Equivalencia con OPC Classic: uso de wrappers en caso de incompatibilidades entre versiones.
- Independencia de plataforma: compatibilidad con todo tipo de plataformas hardware y sistemas operativos.
- Seguridad: asegurar encriptación, autenticación y auditoría, todo manteniéndose compatible con los firewalls existentes.
- Extensibilidad: posibilidad de añadir nuevas características sin afectar las ya existentes.
- Modelado comprensivo de la información: gracias a sus capacidades de orientación a objetos, incluso las estructuras de datos más completas pueden ser modeladas y extendidas.

Lo que nos lleva a poder definir las diferencias entre OPC UA y OPC Classic en cuanto a diversos aspectos:

Aspecto	OPC Classic	OPC UA
Sistemas Operativos	Windows: COM/DCOM	Cualquier SO
Tecno. Embedida	Server en Computadora	Server en PLC
Seguridad	Débil	Cert. de seguridad
Config. Firewall	Muchos puertos	Solo un puerto abierto
Modelo de Info.	Diferentes tecnologías	Tecnología Unificada

Alcance del proyecto

¿Qué pretende el proyecto? ¿Cómo pretende hacerlo? Hay que tener clara cuál va a ser su influencia y las herramientas que se van a utilizar.

El propósito de este proyecto es, mediante el uso de herramientas ampliamente aceptadas como WonderWare, KEPServer o Energy Manager PRO, crear una herramienta transversal. Aunque el enfoque se ponga en industria, y así lo refleje la memoria, la intención es que este tipo de sistemas, dado un lugar donde coexistan diversos dispositivos, y se quiera tener una manera avanzada de controlar lo que consumen, puedan desplegarse para satisfacer esas necesidades. Por lo tanto, de igual manera que serviría para una planta industrial, podría servir para instalaciones agrícolas, o para comunidades de viviendas en zonas con pocos recursos. Por este motivo, este proyecto es trascendente y es aplicable no solo a necesidades actuales y concretas, sino

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

que en un futuro las ideas que plantea serán ampliamente utilizadas y podrían convertirse en paradigmas.

Conceptos generales

Aunque escrita de manera que cualquiera, con un mínimo de conocimiento técnico, pueda entenderla, esta memoria trata un tema “de nicho”, y como tal, consta de algunos conceptos que hay que aclarar, antes de profundizar más en el desarrollo de este proyecto.

Mejora continua

Es en éste concepto donde las compañías ponen más recursos y proyectos de futuro: la idea es contar con un departamento dedicado especialmente a la Mejora Continua, de manera que se utilicen los datos proporcionados por el sistema para realizar un análisis en tiempo real, y así ser capaces tanto de detectar disfuncionalidades en el proceso productivo como de realizar proyectos destinados a solventar las incidencias detectadas, siempre con el propósito de mejorar la eficiencia. Un ejemplo que sirva para entender el concepto es el siguiente:

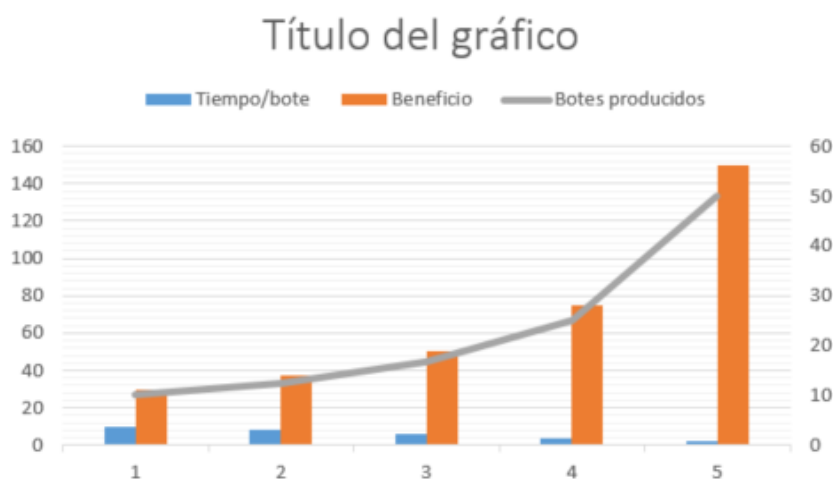


Ilustración 11: Gráfica Mejora Continua

Tiempo total	Tiempo/bote	Coste/bote	Precio venta	Beneficio	Botes producidos
100	10	2	5	30	10
100	8	2	5	37,5	12,5
100	6	2	5	50	16,66666667
100	4	2	5	75	25
100	2	2	5	150	50

Ilustración 12: Datos Mejora Continua

Se puede observar en la tabla de datos como manteniendo el tiempo total del proceso productivo (Tiempo total), el coste por bote (Coste/bote) y el precio de venta por bote (Precio venta), si se mejora la eficiencia del proceso

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

(decrecimiento de Tiempo/bote), se experimenta tanto un incremento de los botes producidos (Botes producidos), como, lógicamente, del beneficio (Beneficio).

El ejemplo anterior, aunque no maneja datos reales, ilustra el propósito del concepto de Mejora Continua. Es el principal motivo del crecimiento de la Industria 4.0, y el mayor argumento que empresas como esta esgrimen a la hora de convencer a potenciales clientes. La idea es sencilla:

- A la empresa A le cuesta 0.5€ la fabricación de un producto, que vende por 2€.
- A la empresa B, gracias a sistemas como los que se han explicado, le cuesta igualmente 0.5€ fabricar un producto equivalente, pero como lo hace de manera más eficiente, produce más unidades, por lo que se puede permitir venderlo por 1,8€.
- B seguiría manteniendo un margen de beneficio razonable, pero satisfacerían las mismas necesidades que A a precio reducido.
- Es lógico pensar que B acabaría desplazando a A del mercado, ya que la primera se ha vuelto más competente.

Esta dinámica obliga a las empresas a ejecutar un análisis de mercado, para ver el estado del segmento en el que compite, y determinar el estado de evolución de sus rivales en relación a Industria 4.0. Posteriormente, tiene que decidir cómo abordar su actualización a estos métodos de producción.

Corrección del factor de potencia

En el mundo de la electricidad, controlar el flujo de energía a través de la instalación, y contabilizar el consumo efectivo que las máquinas de planta producen, no es tan simple como hacer un sumatorio de lo que se supone que consumen los equipos.

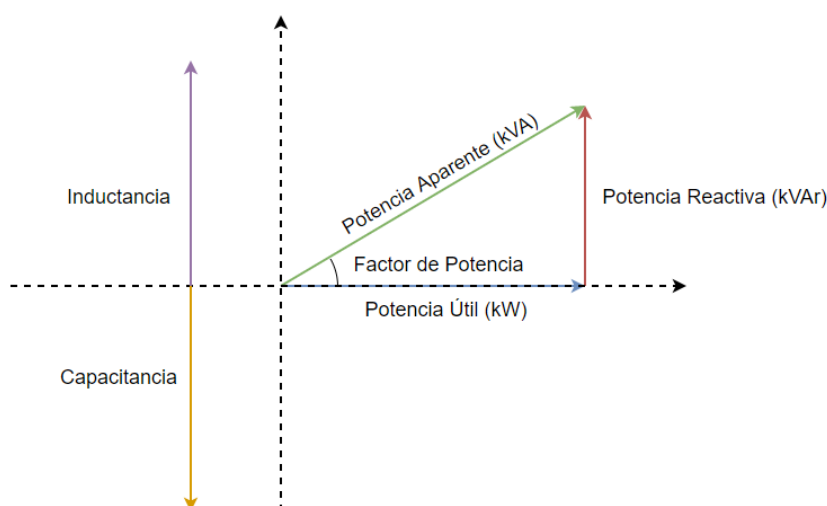


Ilustración 13: Triangulo de Potencias

Si seguimos la teoría, hay tres tipos de potencia:

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

- Útil (kW): Potencia capaz de convertir la energía eléctrica en trabajo. Es la que determina la demanda eléctrica de una instalación.
- Reactiva (kVAr): Potencia utilizada para la formación de los campos tanto eléctrico, como magnético, de los componentes de un circuito de cierta impedancia. Ésta circula entre dichos componentes, y la fuente de energía del circuito.
- Aparente (VA): Suma de las potencias útil y reactiva. Es la energía con la que se cuenta, a efectos prácticos, para realizar trabajo.

Esto plantea una situación que las compañías distribuidoras tienen que considerar, y es que, mientras que un cliente puede consumir una cantidad determinada de energía eléctrica, si no regulan los límites entre los que se puede mover el factor de potencia, es posible que dicho cliente este pagando por menor energía de la que en realidad está utilizando. Esto es, si el factor de potencia, el ángulo del triángulo, crece, quiere decir que la inductancia del circuito también crece, y que, por lo tanto, en el circuito hay mucha más energía de la que se consumió de la red de distribución. En este contexto, las distribuidoras pierden dinero, y obligan a que el coseno del ángulo de mueva en torno a valores de 0,9 y 1, para que la energía que los clientes utilicen sea la que pagan, y no más. Claro está, las industrias juegan con esos límites para beneficiarse de los efectos de la potencia aparente, sin salirse de la legalidad.

Para corregir la inductividad de la instalación, en caso de arriesgarse a exceder los límites legales, se tendrían que instalar equipos de capacitancia, como baterías de condensadores, para corregir el factor de potencia.

Tales decisiones solo se pueden tomar si se sigue un proceso de Mejora Continua, iniciado en la instalación de sistemas de monitorización como el propuesto en éste proyecto.

Evasión de picos de demanda

Además de los equipos encargados de realizar las tareas productivas de una compañía, existen toda una serie de instalaciones, que también pueden llegar a consumir una cantidad notable de energía, como el aire acondicionado, o el alumbrado de las plantas.

En el caso de los domicilios particulares, si la demanda en un instante excede el máximo contratado, se activan protecciones, configuradas por la propia distribuidora, hasta que la demanda baje a los límites contratados. No obstante, siempre se puede cambiar el máximo contratado contactando con la distribuidora.

En industria, esto es diferente. Si una compañía excede su máximo contratado, la distribuidora impone una multa notable, ya que considera que el exceso que se ha demandado, ha tenido que ser sustraído del suministro de otro cliente.

Para evitar estos problemas, la industria cuenta con la posibilidad de utilizar sistemas de automatización, de manera que se pueda controlar qué está encendido en cada momento. Por ejemplo, es posible que no interese poner en marcha, de manera simultánea, las máquinas de producción, y los sistemas de ventilación. En tanto que ambos utilizan motores, tienen una demanda inicial

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

notablemente mayor a su consumo en régimen de trabajo, ya que tienen que ejercer inercia en los mecanismos. Para estos casos, una decisión inteligente sería, por ejemplo, inicializar las máquinas, y una vez la demanda bajase, activar la ventilación.

Automatización: definición y tipos

Si se va a tratar con este concepto a lo largo de la memoria del proyecto, hay que aclarar su significado y distinguir entre sus diferentes tipos.

Partiremos de un ejemplo para introducir la idea:

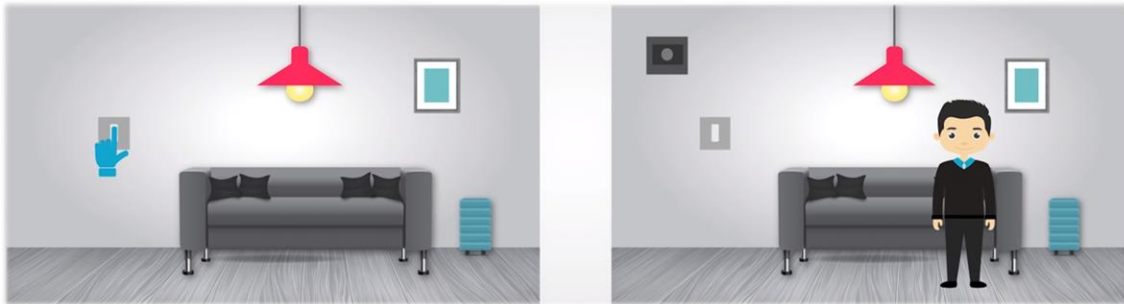


Ilustración 14: Ejemplo Automatización

Si entendemos el encender la luz la habitación como un proceso, podemos diferenciar dos maneras diferentes de abordarlo, según la imagen. A la izquierda, hay que pulsar el interruptor para cerrar el circuito y que la luz se encienda. Esta vía deja la puerta abierta a malgastos energéticos, y puede resultar poco conveniente tener que desplazarse siempre al interruptor para alterar la iluminación de la estancia. Es lo que llamaríamos proceso manual.

A la derecha tenemos un sensor que detecta cuándo hay alguien en la habitación y cierra el circuito de manera autónoma. De la misma manera, cuando no detecta movimiento en un cierto tiempo, abre el circuito y la habitación queda a oscuras. Esto minimiza el riesgo de tener la luz encendida innecesariamente, y libera el tiempo invertido en ir hacia el interruptor para llevar a cabo otras tareas.

Se intuye entonces lo que aporta la automatización a la industria. Una empresa pequeña puede satisfacer a sus clientes elaborando los productos a mano, y mantener una ratio tiempo invertido-calidad razonable, pero ésto es falso una vez la compañía crece. Cuando la masa de clientes crece, no resulta factible intentar producir un mayor número de productos y mantener la misma ratio de antes: habrá que sacrificar calidad para cumplir los plazos acordados. Además, los humanos cometen errores que una máquina programada difícilmente podrá cometer.

Sabiendo esto, podemos pasar a diferencias entre los distintos tipos de automatización:

Automatización Discreta

Es el proceso por el cual se fabrican productos cuantificables, es decir: podemos contar cuántas unidades se han creado. Esto permite detener la producción después de la fabricación de una unidad y retomarla posteriormente sin más implicaciones en cuanto al proceso (hay que tener en cuenta la naturaleza específica de lo que se está produciendo).

El potencial de este tipo de automatización reside en la composición de diferentes procesos, es decir, el uso del producto de un proceso como la entrada de otro. Permite crear partes complejas que posteriormente pueden pasar a formar parte de otra entidad que las agrega.

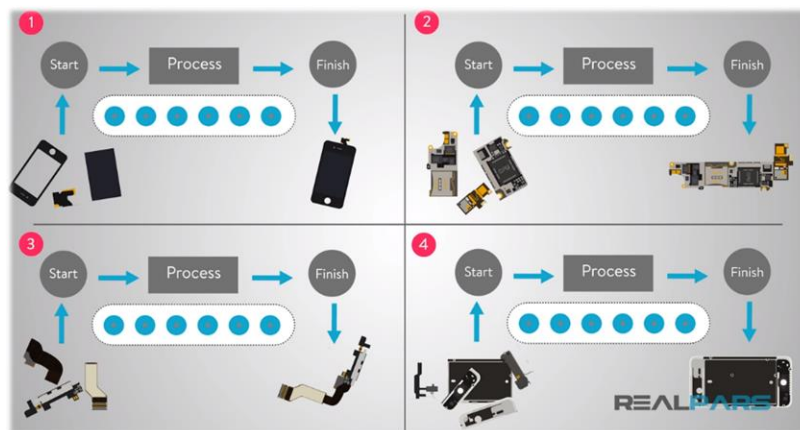


Ilustración 15: Automatización Discreta

Automatización Analógica

Al contrario que con la automatización discreta, el tipo analógico no admite paradas de manera tan simple, ni su producto es tan fácilmente medible. El proceso de fabricación analógico requiere que las materias primas se traten y mezclen para crear el producto final, mientras que en el discreto la fabricación no tiene por qué ser llevada a cabo por un solo proceso. Un ejemplo sería la fabricación de helado.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

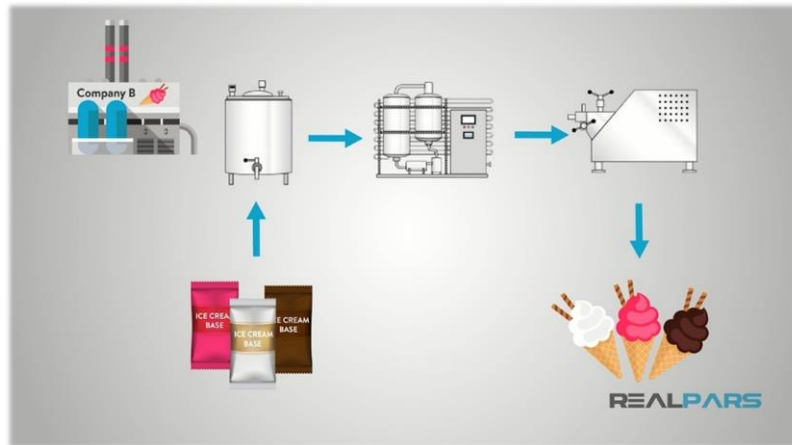


Ilustración 16: Automatización Analógica

Una compañía A podría mezclar nata, huevos y azúcar para hacer la base del helado, para luego enviarlo a la compañía B, que combinaría dicha base con sabores y añadiría otros componentes como, por ejemplo, galleta.

OEE

Siglas para Overall Equipment Effectiveness. Es una medida de cómo de bien funcionan ciertos recursos comparado con su rendimiento teórico máximo, de manera que el ideal sería que todo el equipo de una planta rindiese con una OEE del 100%, aunque eso es virtualmente imposible.

El cálculo porcentual del OEE depende de otros tres porcentajes: disponibilidad (availability), rendimiento (performance) y calidad (quality). La operación sería la siguiente:

$$OEE = Availability * Performance * Quality$$

Donde la disponibilidad indica qué tanto por ciento del tiempo la máquina en cuestión ha estado dispuesta para producir, el rendimiento marca cuánto ha producido respecto a lo que se tenía planeado, y la calidad muestra cuántos de los productos producidos pasan el control de calidad.

En conclusión, se puede entender OEE como una medida general del trabajo productivo, pero siempre teniendo en cuenta que depende de factores que vienen directamente relacionados con el comportamiento de las partes que conforman el proceso productivo. Si se tiene claro esto, se puede ver cómo encaja el análisis de los datos brindados por el MES en el proceso de la mejora del OEE.

$$MES \rightarrow Mejora Continua \rightarrow Incremento del OEE$$

KPI

Siglas para Key Performance Indicator. En general, un KPI es un indicador de éxito relacionado con cierta actividad, usado para evaluar cómo de bien se está llevando a cabo, si se tienen en cuenta los objetivos preestablecidos.

Si nos centramos en el rol que juegan los KPI en un sistema industrial, encontramos algunos como las unidades buenas producidas, las malas, los tiempos de paro y de ocio, el tiempo total de producción, y otros indicadores más relacionados con el consumo de recursos, como el agua y/o gas utilizados, la energía eléctrica consumida, etc.

Relación KPI-OEE

El objetivo final del proceso de mejora continua es el de incrementar el OEE. Éste es el marcador final que dicta el éxito que ha tenido la implantación de un sistema de automatización, pero hay que tener clara la relación que los KPI definidos en nuestro proceso tienen con el OEE final. Podríamos ver cada uno de las tres variables que forman la ecuación de la OEE como el producto de operar con ciertos KPIs:

$$\text{Availability} = \text{Tiempo productivo} / \text{Tiempo total}$$

$$\text{Performance} = \text{Unidades producidas} / \text{Unidades planeadas}$$

$$\text{Quality} = \text{Unidades producidas buenas} / \text{Unidades producidas}$$

Donde cada una de las variables que entran en juego son KPIs del sistema, y además sirven para encontrar el valor de otros (Unidades producidas malas = Unidades producidas - Unidades producidas buenas).

ERP

Siglas para Enterprise Resource Planning. Es un sistema de gestión de procesos comerciales. Si se comunica con el MES, gana la capacidad de manejar datos en tiempo real, permitiendo automatizar algunas operaciones financieras y productivas que dependen de las lecturas que haga del sistema. En una industria automatizada, el ERP se encarga de gestionar todos los recursos de la compañía, ya sea dinero, o stock en almacén, y monitoriza todo lo relativo a consumo del proceso productivo.

De esta manera, la automatización trasciende de la propia planta, y hace que el MES sea capaz de ordenar tareas como dar orden a un proveedor para que prepare un pedido, o avisar al departamento de logística de que cierto pedido está preparado para ser enviado al cliente.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES



Ilustración 17: ERP

MES

Siglas para Manufacturing Execution System. Una primera aproximación a la definición del MES sería pensar que es algo análogo al Sistema Operativo de cualquier computadora personal. Para acceder a los datos que la máquina guarda, y poder utilizar las aplicaciones de sus recursos, no es realista intentar actuar directamente contra el Hardware, ordenándole que ejecute cierto opcode, con ciertos registros, en una dirección de memoria determinada, por ejemplo. Se necesita una capa intermedia, una interfaz entre los recursos físicos que nos brinda la computadora y cualquier pieza de software que haya en capas superiores.

Si pensamos en una planta de producción industrial, podemos aplicar la misma filosofía. No es realista querer tomar lecturas del proceso productivo a mano, configurar los parámetros de las máquinas desde cero cada vez, o intentar solucionar algún evento de manera inmediata si se depende únicamente del capital humano. Es por eso que hace falta una manera de gestionar ciertos aspectos de la actividad industrial de manera informática, y de brindar al personal de la compañía la posibilidad de visualizar lo que sucede en la planta, tanto en tiempo real como a través de datos históricos.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

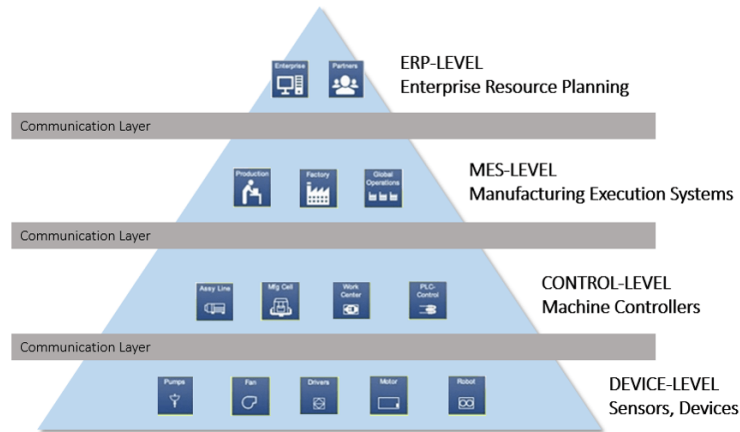


Ilustración 18: Pirámide MES

Un sistema MES, por tanto, sirve para monitorizar el rendimiento de un proceso industrial, y así poder tomar decisiones que permitan mejorar el OEE. Mediante los datos que maneja, se pueden programar alarmas si cierto KPI ha entrado en un rango de valores determinado, o generar reportes que muestren la evolución de una serie de datos.

SCADA

Siglas para Supervisory Control and Data Acquisition. Es un sistema de control industrial que utiliza todo un conjunto de computadoras, redes, PLCs e interfaces gráficas para controlar de manera activa el proceso productivo de una industria.

En cuanto a arquitectura, el SCADA se situaría por debajo de los ordenadores de coordinación, que reciben los planes de producción desde los niveles superiores, y por encima de los controladores de las máquinas de planta.

Se podría decir, entonces, que un SCADA es un intermediario entre los intereses en cuanto a scheduling de la producción, y las máquinas que se tienen que encargar de llevarla a cabo.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

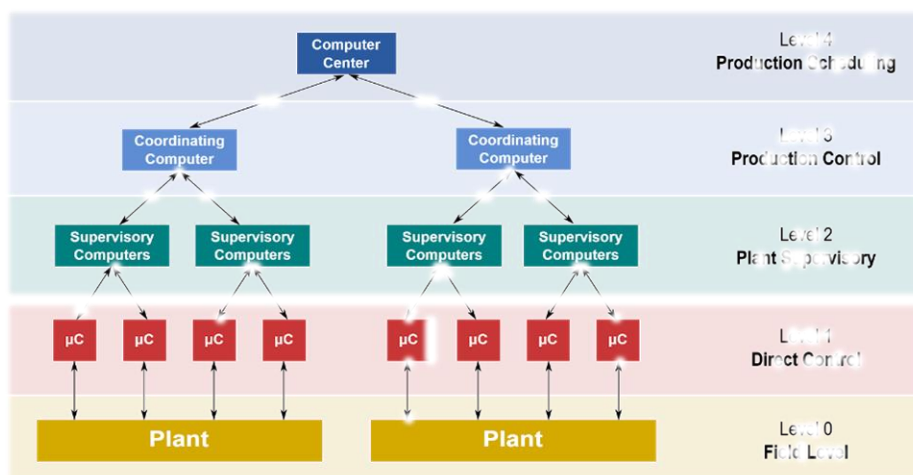


Ilustración 19: Niveles de los Sistemas

SCADA vs MES

Una vez han quedado definidos los conceptos de SCADA y MES, es factible ver ciertas similitudes entre ambos. Ambos monitorizan el proceso productivo, con el propósito de monitorizarlo, pero hay algunas diferencias fundamentales. Mientras que el SCADA consta de actuadores para influir de manera activa en planta, el MES se encarga de las alarmas derivadas de las lecturas recibidas de la misma, y de historizarlas para su posterior análisis en el proceso de Mejora Continua.

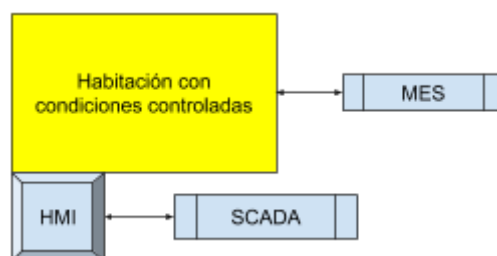


Ilustración 20: SCADA vs MES

Supongamos que tenemos una habitación donde se fabrican productos que necesitan toda una serie de condiciones ambientales (temperatura, humedad), como los medicamentos. Es lógico pensar que tendremos un HMI encargado de darnos acceso a cualquier tipo de datos o acciones que se puedan llevar a cabo sobre esa habitación, así como un SCADA que controle el proceso a nivel general, y un MES que historice las lecturas de los sensores.

Si se detectase que alguno de los parámetros controlados dentro de la habitación ha entrado en un rango inválido, tanto el MES como el SCADA lo sabrían, pero la diferencia es que mientras que el MES guardará en base de datos el momento y motivo por el cual la producción se ha tenido que detener,

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

el SCADA se encargará de hacer parar físicamente la producción, parando las máquinas.

Es en este punto cuando se aprecia que MES y SCADA son partes complementarias de un sistema más grande, y que la completa automatización y Mejora Continua de una industria pasa por la correcta coordinación de las labores de ambos.

HMI

Siglas para Human-Machine Interface. Hace referencia al concepto de poner una capa entre el operario que va a acceder a los recursos de cierta máquina en particular, y el sistema interno de la propia máquina. En el ámbito industrial, un HMI es una computadora conectada a una máquina en concreto, con el objetivo de controlarla y monitorizar su funcionamiento.



Ilustración 21: Human-Machine Interface

Si se tiene una imagen panorámica de todo el sistema, se podría decir que un HMI es a una máquina lo que el SCADA es a todo el proceso productivo. De esta manera, un HMI puede tomar decisiones sobre puntos particulares, mientras que el SCADA puede actuar sobre todo el sistema.

Autómata (PLC)

También conocido por sus siglas en inglés PLC (Programmable Logic Controller). Es una computadora digital, adaptada para controlar procesos productivos, tales como la gestión de líneas de ensamblaje o el manejo programado de los dispositivos robóticos de que disponga la compañía. Para su función, se les exige que sean físicamente resistentes a las condiciones potencialmente adversas que la máquina pueda experimentar, además de brindar una interfaz de programación conveniente.

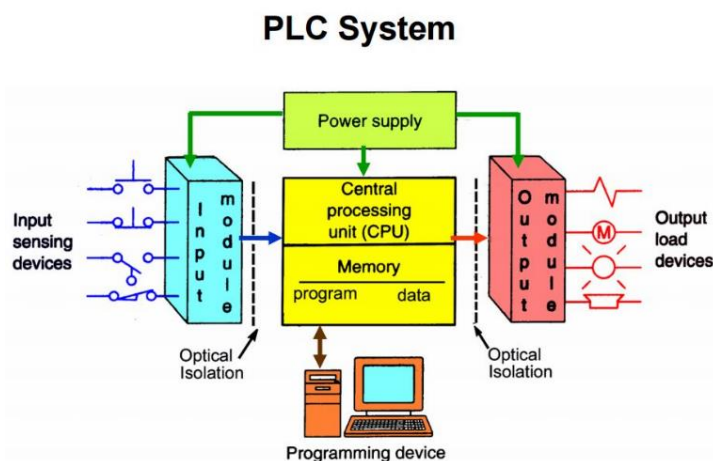


Ilustración 22: Programmable Logic Controller

Lo crítico del trabajo que desempeñan hace que el rendimiento exigido a estos dispositivos se adecue a los requisitos de un sistema de Hard Real Time. Esto quiere decir que el proceso de los datos recibidos y el retorno de los resultados debe realizarse dentro de una ventana temporal muy estricta, ya que de ello depende el rendimiento de cualquier máquina que dependa de las decisiones que tome el PLC.

Para la comunicación, confían en Modbus, el standard de facto en este tipo de comunicaciones. Es un protocolo que resulta fácil de mantener, y maneja bits de manera raw, sin imponer demasiadas restricciones.

Historian

Parte del sistema informático encargada de recoger las lecturas provenientes de los sensores. Para ello tiene que contar con una base de datos capaz de procesar grandes cantidades de datos en poco tiempo, y de clasificar estos convenientemente. Comúnmente se usa el término "tag" para referirse a cada una de las entidades que maneja la base de datos. Un tag, por ejemplo, puede ser la velocidad de una máquina, o un booleano que indique si en ese momento se encuentra realizando alguna tarea, entre otros.



Ilustración 23: WonderWare Historian

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Wonderware

A través de ArchestrA IDE, se definen los mapeos necesarios de las variables leídas en planta y todos los scripts y alarmas necesarios para asegurar que los datos suben al MES de manera conveniente.

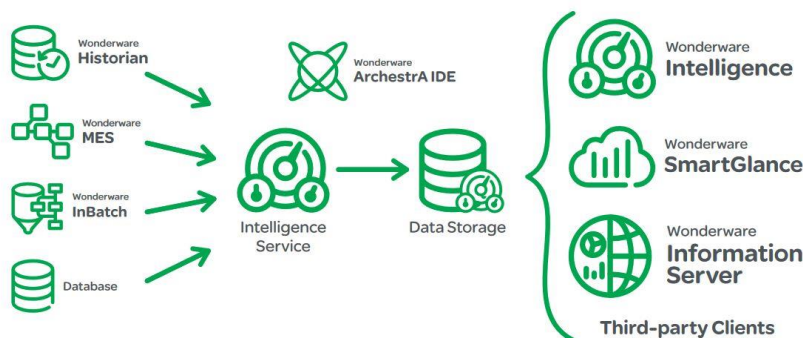


Ilustración 24: WonderWare (ArchestrA IDE)

Wonderware ofrece su propio MES, pero en este caso la compañía desarrolló uno propio, por cuestiones de competencia, y en vez de usar los drivers que incluye el software para poder comunicarse con las máquinas, escogieron una aplicación de terceros que les resultaba más adecuada.

KEPServer

Software encargado de alojar los drivers necesarios para la comunicación de ArchestrA con las máquinas. Se encarga de mapear las señales provenientes de los autómatas a variables, cuya referencia es usada por ArchestrA para hacer su propio mapeo y poner a disposición de los desarrolladores los valores leídos en variables que se puedan manejar. También cuenta con toda una serie de funcionalidades relativas a alarmas y eventos que permiten personalizar el comportamiento del sistema dada una serie de valores, además de una serie de drivers para los fabricantes más comunes, así como paquetes adicionales disponibles bajo demanda.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES



Ilustración 25: KEPServerEX

OLE for Process Control: qué es y cómo se usa

El protocolo de comunicaciones más utilizado a la hora de interactuar con las máquinas es OPC. Aquí se explicará qué es y se hará un pequeño ejemplo de cómo utilizarlo.

OLE for Process Control (OPC) es un estándar de comunicación industrial que asegura un flujo continuo y confiable de información entre los distintos componentes de un sistema de automatización, sin importar el fabricante de los mismos. Esto se consigue gracias a una serie de especificaciones que definen las interfaces tanto Cliente-Servidor como Servidor-Servidor. Gracias a esta interfaz común, los dispositivos HMI/SCADA son capaces de interactuar con un intermediario capaz de traducir cualquier petición OPC genérica a una petición específica para el dispositivo en cuestión, y viceversa.

Esta facilidad de comunicación permite que los desarrolladores se despreocupen de los pormenores de la comunicación con los dispositivos, de manera que solo tienen que especificar al software que maneja la parte OPC con qué dispositivo se van a comunicar.

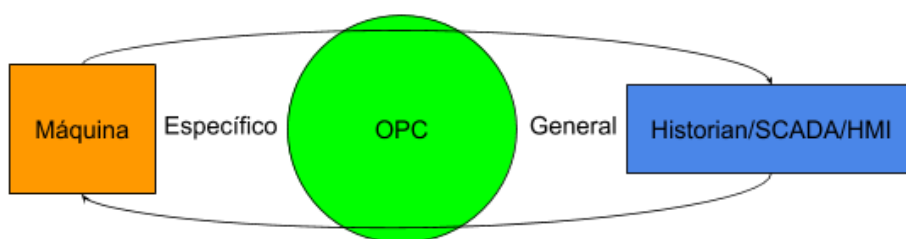


Ilustración 26: Funcionalidad OPC

Hay tres grandes componentes: KEPServer, Wonderware y el MES, y quedarían dispuestos de la siguiente manera:

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

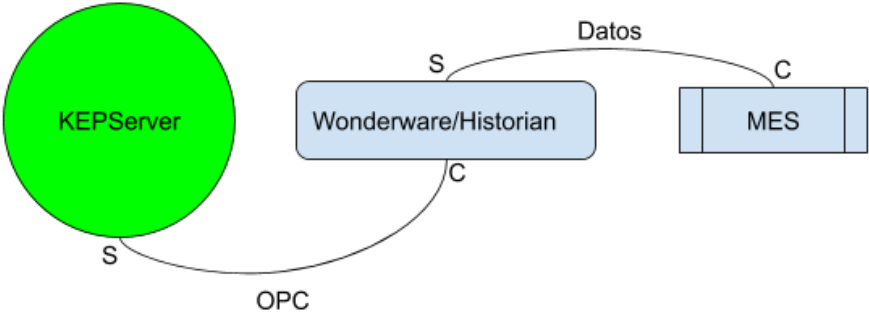


Ilustración 27: Componentes de Comunicación

En la parte OPC, KEPServer hace las veces de servidor de datos, recogiendo los de los autómatas y sirviendolos a Wonderware. De la parte de datos, Historian sirve los datos recogidos al software MES para su análisis a los clientes MES.

Si dejamos de lado la parte del MES, es posible, mediante KEPServer y Arhcestra IDE, realizar una simulación de lo que sería el proceso de tratar señales de los autómatas vía un OPC, y mapearlas con ArchestraA IDE a variables que un desarrollador pueda utilizar para desarrollar aplicaciones.

Después de instalar SQL Server, Wonderware y KEPServer, se tiene todo lo necesario para montar un sistema real, y las opciones de los mismos programas permiten crear simulaciones que emulan el comportamiento de un sistema real ejecutándose.

KEPServer ofrece un dispositivo por defecto, localizado dentro de un canal creado automáticamente, ya preparado para añadir tags según las necesidades presentes:

La imagen muestra la interfaz de configuración de tags en Wonderware. A la izquierda, un árbol de navegación muestra la estructura del proyecto: Project > Connectivity > Channel1 > Device1. A la derecha, una tabla muestra los tags configurados:

Tag Name	Address	Data Type	Scan Rate	Scaling	Description
Tag2	K0001	Word	100	None	Constant Read/Write tag used to verify ...
Tag1	R0001	Word	100	None	Ramping Read/Write tag used to verify ...
Tag3	R0002	Boolean	100	None	

Ilustración 28: Visualización de Tags por Dispositivo

En el desplegable a la izquierda disponemos de una serie de opciones, pero para el caso actual solo interesa la sección Connectivity, y dentro de esta, Channel1 y Device1. Se puede observar cómo se muestran los tags asociados a

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

ese dispositivo, así como el tipo de datos que maneja, el tiempo de escaneo en milisegundos, las opciones de escalado que se le aplican, y una descripción de la información que describe el tag. En cuanto al campo Address, es la dirección desde la cual se puede acceder a los datos en la máquina, y viene dictado por una tabla como la siguiente:

Device Type	Range	Data Type	Access
Registers	R0000-R9999 R0000-R9998 R0000-R9996	Word, Short, BCD DWord, Long, LBCD, Float LLong, QWord, Double, Date, Boolean	Read/Write
Constants	K0000-K9999 K0000-K9998 K0000-K9996	Word, Short, BCD DWord, Long, LBCD, Float LLong, QWord, Double, Date, Boolean	Read/Write
Bits	R0000.00-R9999.15 K0000.00-K9999.15	Boolean	Read/Write
Strings	S000-S999	String	Read/Write

Ilustración 29: Registros de Direcccionamiento

En el propio KEServer, si seleccionamos la opción Quick Client, se nos desplegará, como el nombre indica, un cliente OPC que nos permitirá ver cómo evolucionan los Tags del sistema.

The screenshot shows the 'OPC Quick Client - Sin título' window. On the left is a tree view with 'Channel1.Device1' expanded, showing various data types like 'System', 'Statistics', and 'Examples'. The main area displays a table with the following data:

Item ID	Data Type	Value	Timestamp	Quality	Update Count
Channel1.Device1.Tag1	Word	6352	09:57:13.239	Good	1699
Channel1.Device1.Tag2	Word	1212	09:57:13.239	Good	844
Channel1.Device1.Tag3	Boolean	1	09:57:11.239	Good	422

At the bottom, there is an 'Event' log showing system updates with timestamps and descriptions. The status bar at the bottom right indicates 'Item Count: 301'.

Ilustración 30: Quick Client

Podemos observar cómo se nos muestran los tres tags que vimos anteriormente, el valor que tienen, si se pueden considerar datos buenos, y otra información como el timestamp y cuántas veces se ha actualizado el valor.

Más adelante veremos cómo la información que muestra esta ventana es coherente con la que nos brinda ArchestrA IDE, del cual hablaremos ahora.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

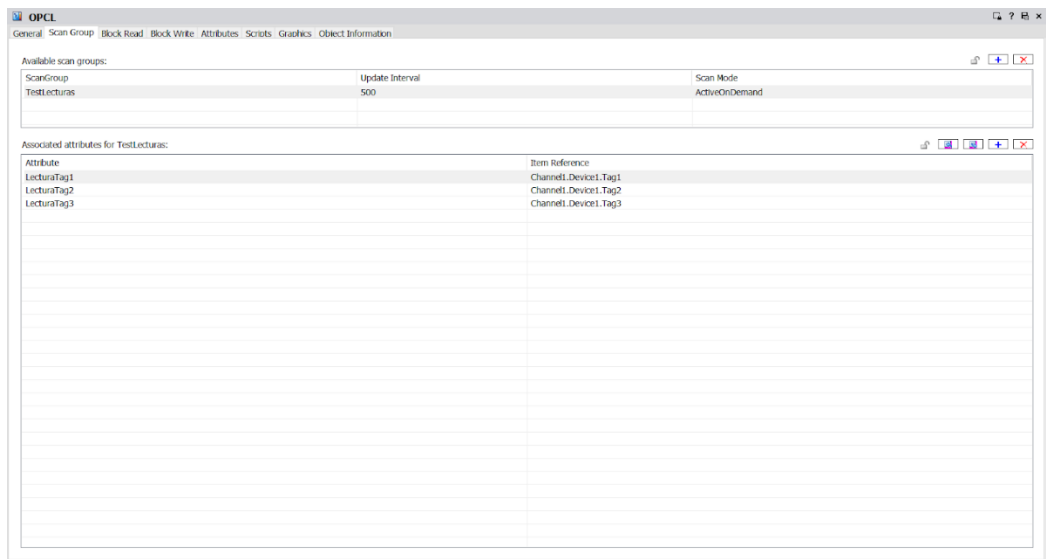


Ilustración 31: Cliente OPC en Archestra IDE

La imagen anterior muestra la sección de Archestra que representa el cliente OPC. Se define un Scan Group, con un periodo de actualización y modo de escaneo determinado, al cual asignar señales provenientes del OPC Server, con el objetivo de mapearlas. Eso mismo es lo que se hace más abajo, donde se asigna un Attribute a cada Item Reference, o lo que es lo mismo: le damos a la señal un nombre para referirnos a ella dentro del programa.

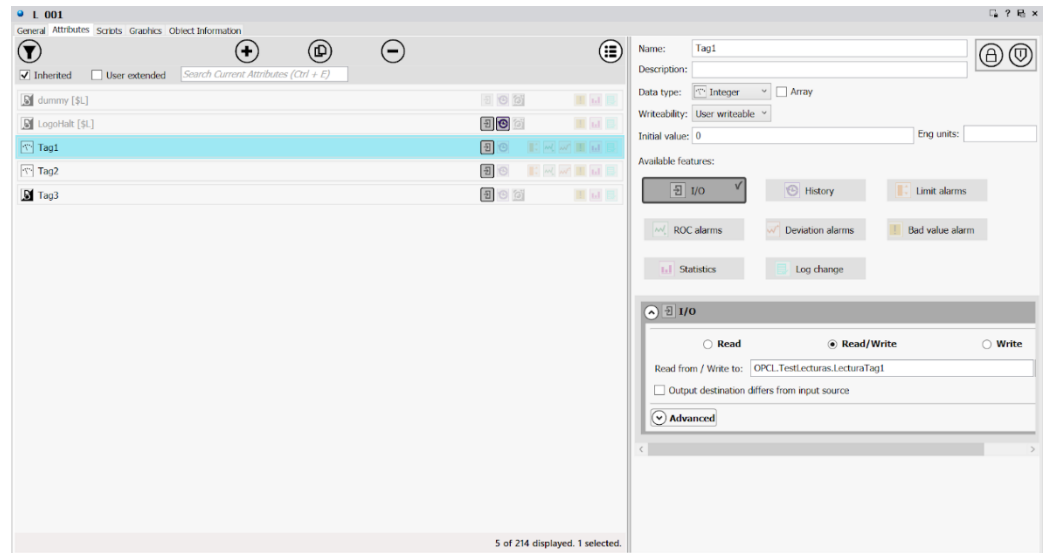


Ilustración 32: Mapeo de Atributos

Como se puede observar, disponemos de tres atributos mapeados con la lectura del cliente OPC vía la feature I/O. De esta manera, hemos llevado los

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

datos de la máquina al Server OPC, de ahí al Cliente OPC, y ahora ya los tenemos disponibles para usarlos según nos convengan. Se ha configurado el tipo de datos y los permisos R/W, pero el resto de características se han dejado por defecto, ya que no eran relevantes para este ejemplo.

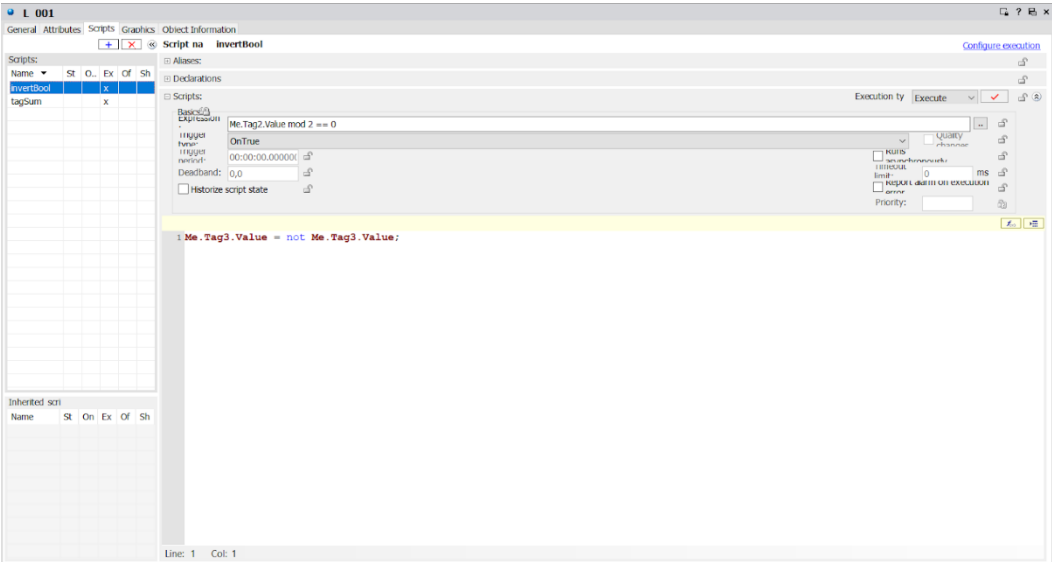


Ilustración 33: Scripting en Visual Basic

Antes de mostrar de manera simultánea la información del Quick Client de KEPServer y la del visor de ArchestraA, es necesario explicar la funcionalidad de scripting incluida en este último.

Gracias a Visual Basic, se pueden elaborar scripts todo lo complejos que uno necesite, con tal de manipular los datos escaneados, y llevar a cabo acciones en función del comportamiento del sistema. Están disponibles librerías de .NET, así como IntelliSense. Se puede escoger ejecutar el script al inicio de la ejecución de sistema, al apagarlo, o como en este caso, al cumplirse cierta condición, pudiendo especificar, además, si se ejecuta una sola vez, o siempre que la condición sea cierta. En este script se quiso que el Value del Tag3 (Boolean), se invirtiera siempre que el Value del Tag2 fuese par.

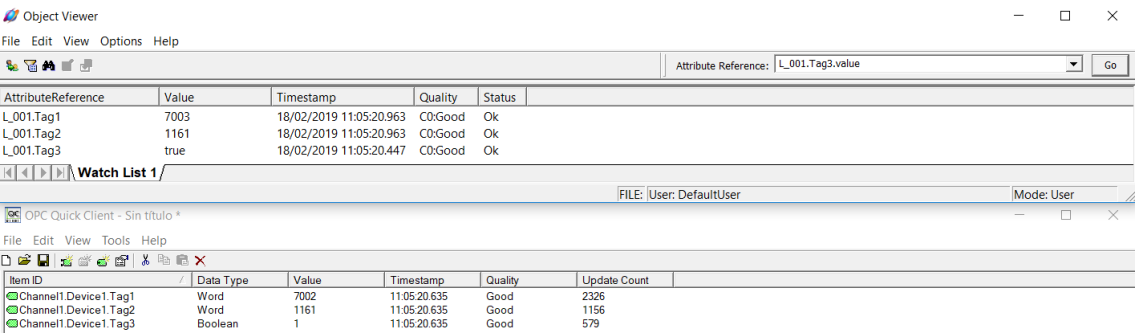


Ilustración 34: Object Viewer

Arriba se muestra el Object Viewer de Archestra, y abajo el Quick Client de KEPServer. Salvando el decalaje que puede haber por los tiempos de escaneo, podemos observar cómo efectivamente los datos se mantienen coherentes en los diferentes programas.

Aunque poco útil, este ejemplo sirve para ilustrar el flujo de datos que hay desde que el automatista extrae las señales, hasta que el desarrollador crea una aplicación.

Real Time y Scan Rate en el OPC

En tanto que un estándar de comunicación industrial pensado para comunicarse con equipos de distinto fabricante y, por lo tanto, con especificaciones y drivers potencialmente distintos, OPC tiene que ser usado mediante un entorno de desarrollo que permita controlar los tiempos con los que el sistema trabaja para pedir y entregar los datos de producción. De esta manera, se pueden sintonizar los tiempos de respuesta para que, dentro de las capacidades del hardware, los equipos se comuniquen con la sensibilidad que requiere el despliegue concreto de la aplicación.

Si suponemos que existe una línea que produce unas decenas de unidades por minuto, y que hay un sensor encargado de detectar el paso de las unidades por un tramo de la línea, para contabilizar la producción, no tiene sentido configurar en el servidor OPC un tiempo de escaneo de milisegundos, ya que leeremos varias veces el mismo valor de la señal del contador de unidades. Por el contrario, si la línea produce cientos de unidades al minuto, sí que es conveniente configurar un tiempo de escaneo más corto, de manera que se sigan produciendo algunas lecturas repetidas, pero se gane en fidelidad a la hora de representar el estado de la línea. El siguiente gráfico ilustra esta idea:

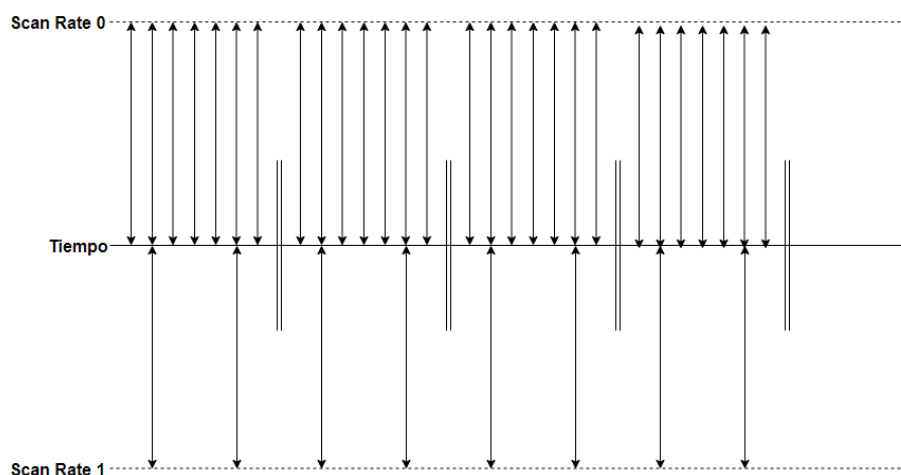


Ilustración 35: Comparación Scan Rates

Si suponemos que las dobles líneas verticales indican los momentos en los que el contador refleja una nueva unidad producida, tenemos que el Scan Rate 0

mantiene siempre actualizado el número de unidades producidas, pero realiza toda una serie de lecturas repetidas que no aportan información nueva, y puede contribuir a la congestión de la red por donde viajan los paquetes. Por otro lado, el Scan Rate 1 también se mantiene actualizado, pero consigue reducir el número de lecturas repetidas. Si además del conteo de unidades producidas, también recogiésemos otros datos, como el rendimiento nominal de la máquina, o su velocidad de funcionamiento, podría ser conveniente un ciclo de lectura similar al de Scan Rate 0, dado que nos mantendría más actualizados en lo referente a todos esos datos.

La idea de los sistemas de automatización industrial es que ofrezcan información en tiempo real, pero hay que sopesar las necesidades reales del sistema, y lo que la infraestructura de redes y equipo nos puede ofrecer. Como aclaración, las tres variantes más comunes del concepto de Real Time son las siguientes:

- **Hard Real-Time:** si no se adquieren los datos antes de un deadline, el sistema falla.
- **Firm Real-Time:** no cumplir el deadline es tolerable, pero degrada la calidad de servicio (Quality of Service), del sistema. Una lectura es inservible si no cumple el deadline.
- **Soft Real-Time:** los incumplimientos del deadline degradan la utilidad de una lectura, pero son tolerados.

Tradicionalmente, cuanto más crítico es el ámbito de trabajo de una determinada industria, más riguroso es el control de los sistemas de tiempo real. No es comparable en cuanto a importancia activar un sistema de frenos ABS a incrementar un contador de unidades producidas en pantalla. Una primera intuición sería elaborar una configuración más ajustada para los sistemas SCADA, ya que tienen que tomar parte activa en el proceso de producción, y parametrizar de manera más laxa el sistema de captación de datos de un MES, pero hay un problema de base en esta idea: ambos sistemas son construidos encima de OPC. Aunque los márgenes de error puedan ser más permisivos en algunos casos que en otros, no es realista pensar en tener una configuración del OPC para cada uno.



Types of Real Time Systems

- **Hard real time systems**
 - Must always meet all deadlines
 - System fails if deadline window is missed
- **Soft real time systems**
 - Must try to meet all deadlines
 - System does not fail if a few deadlines are missed
- **Firm real time systems**
 - Result has no use outside deadline window
 - Tasks that fail are discarded

Ilustración 36: Tipos de Sistemas de Tiempo Real

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Por lo tanto, el método de trabajo más conveniente es ajustar el OPC para que cumpla con las exigencias más estrictas de todas las que se presentan, y retocar a en niveles más altos de la arquitectura algunos parámetros, si para ese punto resulta más conveniente aplicar políticas más laxas.

Organización de la infraestructura: galaxias

Mediante Archestra IDE, se puede gestionar la infraestructura de nuestro sistema, es decir: organizar las señales que se manejan, tomar decisiones sobre qué hacer en ciertas situaciones, etc. Para esto, existe el concepto de galaxia.

Una primera aproximación al concepto de galaxia sería pensar en ella como nuestra aplicación de manera integral. Es la entidad que engloba todos nuestros equipos en red, que aplica una serie de políticas a todos ellos, y que consta de una serie de bases de datos para almacenar la configuración del sistema en cuestión. Para conseguirlo, se dispone de un conjunto de elementos que sirven para crear la jerarquía adecuada en el sistema, y aplicar las configuraciones que convengan. La siguiente captura ilustra una galaxia simple, que servirá para explicar la lógica detrás de su jerarquía, y representa una industria con una sola línea de producción:

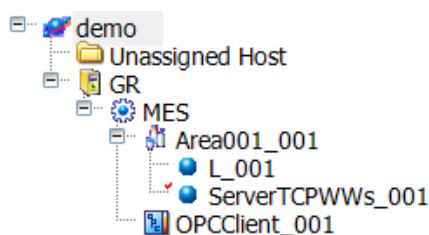


Ilustración 37: Jerarquía de la Galaxia

En el nivel más alto tenemos "demo". Éste representa la galaxia en su conjunto, y sirve para tomar decisiones que afectan a la aplicación entera. Más abajo, la carpeta "Unassigned Host" se crea por defecto, y no tiene uso en este caso.

"GR" hace referencia al WinPlatform donde se despliega la galaxia. Es el nodo encargado de gestionar el inicio, el mantenimiento, y el cierre de esta, y es el centro del conjunto de elementos del sistema.

"MES" es el AppEngine, es decir, el objeto residente en un nodo que se encarga de alojar la ejecución de todo el resto de líneas, dispositivos, y aplicaciones que estén situadas en posiciones inferiores de la jerarquía, y establecer una configuración general para todas ellas. Como se puede observar en la imagen, tiene asociado un cliente OPC, encargado de poner a disposición del desarrollador las señales físicas recogidas en la planta donde se despliega el sistema.

El objeto "Area001_001" representa nuestra única línea de producción. Dentro de ella se podrán crear todos los elementos que permitan monitorizar y tomar decisiones sobre lo que suceda en la propia línea. Es el caso que nos ocupa,

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

“L_001” representa nuestra única línea de producción, y en ella se pueden definir variables, que toman los valores que llegan a través del cliente OPC, y scripts, que permiten tomar decisiones diversas según los datos de que se dispone. Por otro lado, tenemos “ServerTCPWWs_001”, un elemento personalizado por la propia empresa para alimentar por protocolo TCP/IP el sistema de monitorización MES con los datos historizados en WonderWare (es decir, guardados en la base de datos de la galaxia).

Como apunte final, cabe decir que cada nodo tiene asociada una dirección que lo identifica. Es por esto que, si hay varias galaxias, y el nodo en el que se supone que se despliegan es el mismo, solo se nos permite tener desplegada una a la vez, es decir. Si decidimos añadir más sensibilidad al sistema, se pueden añadir varios AppEngine al mismo nodo, de manera que se puedan tomar conjuntos de políticas diferentes según convenga a las distintas zonas de nuestra industria.

Industria 4.0

En tanto que tendencia al alza, y marco tecnológico y filosófico en el que se ubica el proyecto, es necesario definir qué es la Industria 4.0.

Antecedentes

Más allá de lo técnico, es útil conocer el contexto histórico de la evolución de la industria.

En industria, siempre ha sido capital tener el control del proceso productivo, desde las instalaciones y máquinas utilizadas para llevar a cabo la manufactura, hasta los datos que ilustran cómo evoluciona el funcionamiento de la empresa, que muestran posibles errores en el proceso, y hacen posible una acción focalizada que permita mejorar el rendimiento.

A finales del siglo XVIII, la introducción de maquinaria propulsada por agua o vapor conllevó un aumento de la productividad industrial, además de la introducción de fuentes de energía antaño no consideradas, que necesitaban de una gestión especializada. Las nuevas máquinas hacían el trabajo más rápido, pero podían fallar, y en tal caso habría que poder determinar qué causó el fallo, y ver en qué medida era evitable. De manera similar, era necesaria una infraestructura que pusiese las nuevas fuentes de energía a disposición de la industria. Se evidenció entonces la necesidad de una monitorización eficaz del proceso industrial, y se empezaba a atisbar lo que actualmente se conoce como Mejora Continua.

Más tarde, la introducción de la electricidad y del concepto de cadena de montaje siguieron haciendo crecer la complejidad de la industria, pero fueron las siguientes revoluciones las que cambiarían el paradigma según el cual se monitorizaba y gestionaba el proceso productivo, primero a mediados del siglo XX y más tarde en la actualidad, en lo que se conoce como la cuarta revolución industrial.

El siguiente gráfico ilustra, a grandes rasgos, qué introdujeron las tres revoluciones industriales anteriores, y qué está introduciendo esta cuarta:

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

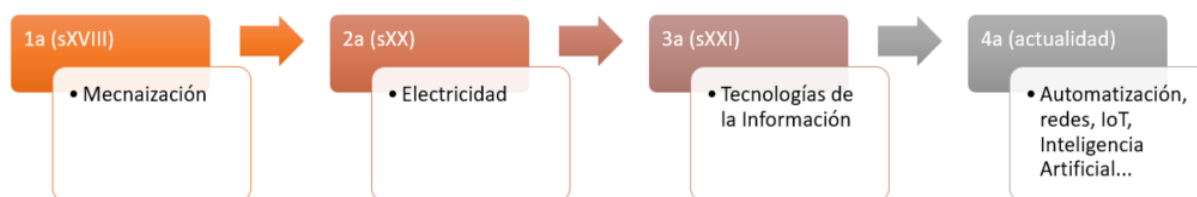


Ilustración 38: Revoluciones

En la actualidad, la tendencia a la automatización y al intercambio de datos en el proceso industrial hacen que sea necesaria una manera rápida y eficaz de saber lo que está pasando, hasta el punto de saber en tiempo real el estado de las líneas de producción. Es aquí donde entran en juego los sistemas MES, y toda la infraestructura, tanto Software como Hardware, que hacen posible que funcione. Como el gráfico muestra, sólo se puede intuir hasta qué punto las nuevas tecnologías penetrarán en el ámbito industrial. Hay un gran número de tecnologías que podrían encontrar entre sus aplicaciones un nicho en industria. Es el caso, por ejemplo, de las tecnologías de construcción de redes y la visión por computador, entre otras.

Un nuevo paradigma de industria: tecnología que lo sustenta

No es difícil deducir las novedades tecnológicas que propulsaron la evolución industrial en el pasado, pero ¿cuáles la propulsarán en el futuro?

Es importante tener una idea intuitiva de lo que supone esta nueva revolución industrial que se está llevando a cabo en la actualidad: de dónde viene y qué ofrece. Pero es, desde un punto de vista técnico, igual de importante saber los pilares sobre los que se está construyendo esta revolución, es decir, las tecnologías que la hacen posible.

Big Data y Analytics

En un contexto en el cual hay grandes cantidades de datos por analizar, información clave por extraer, y decisiones y análisis que realizar en tiempo real, se necesitan medios a través de los cuales poder manipular de manera conveniente la información que se trata, y sacar el máximo provecho de ella. Es por eso que el Big Data Analytics es tan útil: permite aplicar y encontrar patrones, aplicar tendencias de mercado y de los clientes para moldear la nube de bits que fluye a través de los grandes sistemas actuales.



Ilustración 39: Big Data and Analytics

Robots autónomos

Cuando se utiliza el término “robot”, se suele utilizar, sobretodo en la cultura popular, a una máquina de aspecto humanoide. Nada más lejos de la realidad, en nuestro día a día entramos en contacto con multitud de robots, sin que estos tengan aspecto humano, y nos ayudan a realizar ciertas tareas. Entonces se presenta claro que ámbitos concretos de la sociedad, como en este caso la industria, también han encontrado una aplicación para estas tecnologías.

La introducción de las Tecnologías de la Información comportó el uso de máquinas programables, capaces de realizar tareas de manera más rápida y precisa que un operario humano. En el futuro, se espera no solo la presencia de estos robots, sino que puedan trabajar y comunicarse junto a otros robots y operarios, y puedan aprender a realizar sus tareas mejor.



Ilustración 40: Coworking

Simulación

En entornos de trabajo cada vez más complejos, se necesitan herramientas que permitan tener una idea lo más exacta posible de cómo se llevará a cabo un determinado proceso una vez se inicie en la vida real. Para este propósito

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

existen las herramientas de simulación, que permiten, por ejemplo, construir un modelo de una línea de producción, de manera que se pueden llevar a cabo procesos completos, y analizar los resultados antes de empezar efectivamente con la producción.



Ilustración 41: Simulación Virtual de Planta

Integración horizontal y vertical de sistemas

En una industria automatizada, se tiene que luchar por conseguir que ésta sea un ente cohesionado, en el que exista un alto grado de comunicación interno. Para ello, se tiene que asegurar tanto la comunicación entre partes de un mismo nivel de la compañía (intercomunicar los robots de producción), como el flujo de datos entre distintos niveles (las lecturas de planta influyen en el departamento de compras).

Internet of Things Industrial

La proliferación de los microcontroladores y los sistemas embebidos hacen que partes hasta ahora aisladas del proceso productivo puedan comunicarse con otros equipos, y con sistemas de control. Estos elementos abren también la posibilidad a un proceso de datos descentralizado, que aproveche el paralelismo y capacidad de los dispositivos de la red, y facilite la toma de decisiones en tiempo real.

Esta tecnología es la evolución natural de los Distributed Control System, donde el uso de la nube y la computación distribuida hacen posibles grados más altos de automatización que anteriores paradigmas de control industrial.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

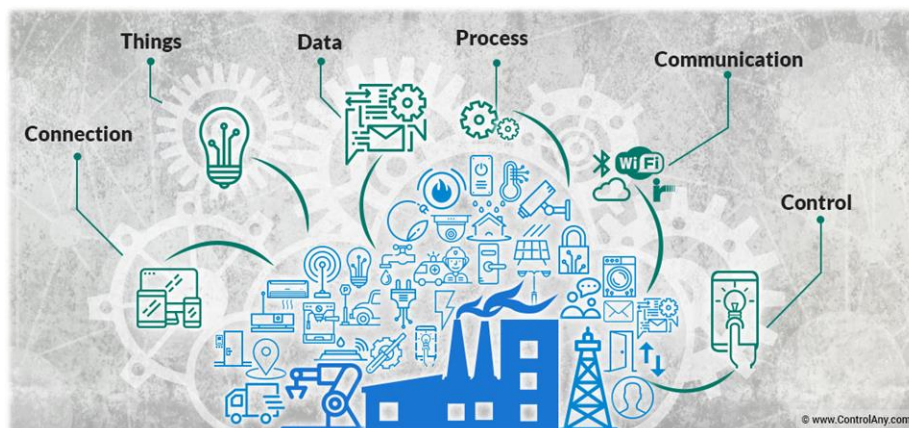


Ilustración 42: Industrial Internet of Things

Cybersecurity

El uso de redes no es nuevo en industria, dado que cualquier planta que utilice algún tipo de equipo informático, por simple que sea, necesita identificarlo y poder comunicarlo con el servidor de datos de que disponga la compañía. No hay que olvidar que, aunque las redes industriales utilizan equipos específicos, y tienden a ser estancas, con la entrada de la industria 4.0 el grado de interconectividad aumenta, y la red interna se puede abrir al exterior, por lo que nacen nuevas brechas de seguridad que hay que tener en cuenta.



Ilustración 43: Cybersecurity en Industria

El Cloud

El crecimiento en potencia de proceso y almacenamiento de la nube hacen que partes del negocio se replanteen para aprovechar lo que ofrece. Esto lleva a realizar cálculos y almacenamiento de manera remota, e incluso a confiar en sistemas SaaS, donde la compañía que ofrece el servicio ofrece una dirección IP, desde la cual se puede visualizar y manipular la información almacenada en los servidores remotos.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

El uso de la nube va estrechamente relacionado con una correcta implantación de políticas en lo tocante a ciberseguridad, y es una muestra más de que, aunque la inclusión e integración de tecnologías comportan beneficios notables, no hay que olvidar que conllevan potenciales fuentes de errores y fallas de seguridad que hay que saber tratar.

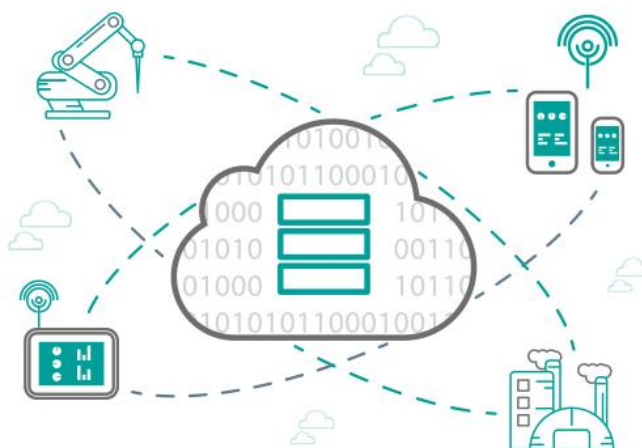


Ilustración 44: Cloud en Industria

Fabricación aditiva

La introducción de las impresoras 3D en el flujo de trabajo de las corporaciones hace que sea plausible la producción de pequeños lotes de componentes o prototipos, con características de construcción especiales. El rápido avance de la impresión 3D permite crear partes complejas, con engranajes móviles, diferentes colores e incluso diferentes propiedades de material.

Para entender hasta qué punto la impresión 3D puede llegar a revolucionar el prototipado y la fabricación de piezas complejas, recurriremos al concepto de voxel que utiliza HP en sus impresoras de última generación.



Ilustración 45: Manufactura Aditiva

De la misma manera que la unidad mínima de visualización es el píxel, al cual se le asigna cada cierto momento un color, en impresión 3D la unidad mínima es el voxel. Si se hace la analogía, un voxel es a un objeto lo que un píxel a una imagen, siempre teniendo en cuenta que se está pasando de un entorno 2D a uno en 3D. Con esto claro, los sistemas modernos de impresión son capaces de controlar el color, textura y material de cada voxel, y hacen posible crear, por ejemplo, un objeto que en parte es liso, rojo y flexible, mientras que el resto es blanco, rugoso y rígido. También se pueden especificar otras características, como las propiedades eléctricas, la opacidad, la elasticidad o la fuerza, pudiendo crear nuevos materiales sintéticos hasta ahora no utilizados.



Ilustración 46: Propiedades de Voxel

Realidad aumentada

Aunque en su infancia, estos sistemas permitirán llevar a cabo acciones a distancia, como marcar un objeto del almacén o interactuar con diferentes sistemas. Los sistemas de visualización harán que los operarios dispongan de información en tiempo real, y agilizarán las comunicaciones entre miembros de la compañía.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES



Ilustración 47: Realidad Aumentada

ODBC

Open DataBase Connectivity es una API destinada al acceso de sistemas de gestión de bases de datos. La justificación de ODBC es que está pensada para ser independiente tanto del sistema de base de datos como del sistema operativo donde se ejecute, por lo que las aplicaciones que utilizan esta tecnología pueden ser llevadas a cualquier plataforma con solo algunos cambios en el código.

El funcionamiento de ODBC se basa en el uso de drivers, que se sitúan entre la aplicación y el gestor de bases de datos, de manera análoga a cómo los sistemas operativos interactúan con los periféricos. La aplicación utiliza funciones ODBC a través del driver, y éste transmite la consulta al gestor de bases de datos.

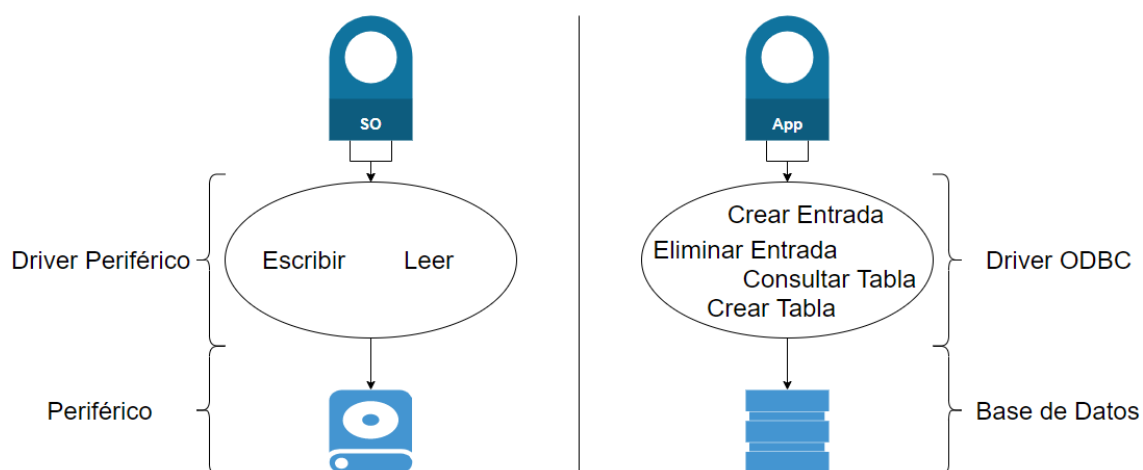


Ilustración 48: Analogía Periféricos-ODBC

Web Services

En esencia, un Web Service es una comunicación entre dos máquinas. En su forma más simple, una recibe los datos que la otra le transmite, normalmente en formatos interpretables por las máquinas, como JSON o XML, a través de protocolos de red como HTTP.

Siguiendo esta filosofía, es común que los sitios web ofrezcan, mediante servicios, los mismos datos que muestran en sus páginas, creando interfaces intuitivas para quien quiera consumir la información. Éstas son la puerta de acceso a los servidores de quien aloja el servicio, y es el lugar al cual cualquier cliente (server, aplicación, etc...) se tiene que dirigir para obtener los datos.

En cuanto a la lógica de funcionamiento de los Web Service, hay varios actores que entran en juego:

- Requester: equipo que quiere consumir la información.
- Provider: equipo que ofrece la información:
- Broker: pone en contacto al Requester con un Provider que ofrezca el tipo de información que busca.

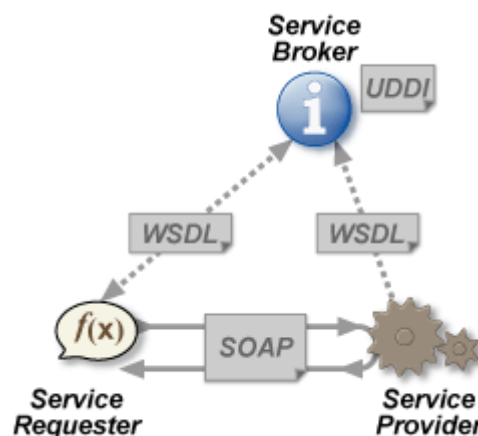


Ilustración 49: Funcionamiento Web Services

El Provider envía un mensaje WSDL (Web Services Description Language) al UDDI (Universal Description, Discovery and Integration) para que los servicios que ofrece queden registrados, y sean visibles en Internet. Posteriormente, los Requester se comunicarán con el UDDI al hacer una consulta, y éste les informará del Provider con quien deben comunicarse. Por último, el Requester se comunica mediante SOAP (Simple Object Access Protocol) con el Provider para pedirle la información, y éste se la entrega en XML.

Integración de Tecnologías

¿Cuál es la imagen general de las tecnologías que intervienen?

Los Manufacturing Execution System son sistemas encargados de, entre otras tareas, automatizar, monitorizar y gestionar todo lo relacionado con el proceso productivo, ya sea la programación de una tarea, la gestión de los recursos

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

utilizados, o las causas de un paro en la producción. Para ello, existen diferentes soluciones, que proporcionan una serie de servicios base, con opción a ampliarlos vía la inserción de módulos.

Algunas compañías desarrollan su propio sistema por completo, mientras que otras optan por utilizar herramientas de terceros, o combinar ambas opciones. Un esquema básico de la implantación de dichos sistemas sería el siguiente:

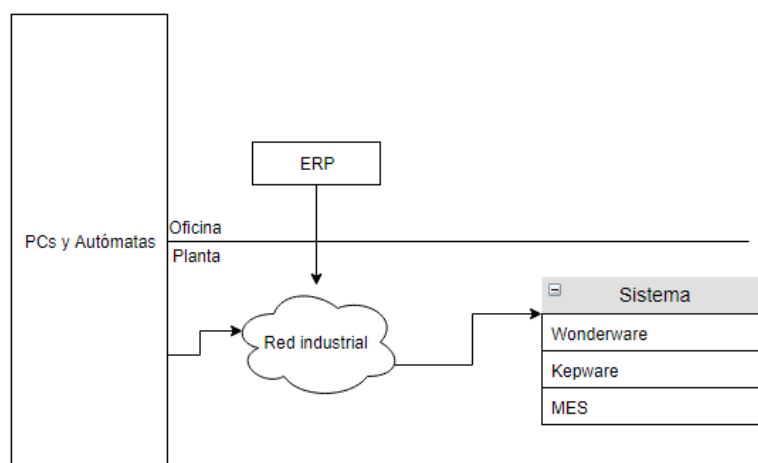


Ilustración 50: Panorama de un Sistema Industrial

A partir de este esquema se pueden deducir las diferencias entre una red comercial y una red industrial. En ésta última (la que nos ocupa), los hosts no son solo PCs, sino que también incluyen a las computadoras que dirigen las máquinas, es decir, los autómatas. En cuanto a arquitectura, este tipo de redes no suelen incluir equipos como firewalls, ya que podría haber colisiones en las direcciones IP, y sería una fuente de problemas adicional que no acaba de estar justificada, teniendo en cuenta la complejidad intrínseca del propio sistema.

Del lado de la oficina, los ERP (Enterprise Resource Planning) se encargan de registrar encargos por parte del equipo comercial, y de enviarlos al sistema en forma de Orden de Producción. En planta, se recogen cantidades masivas de datos de los autómatas, y se envían a través de la red al sistema MES. En estos casos, las bases de datos escogidas para recibir las lecturas de las máquinas suelen ser del tipo no relacional, ya que son más adecuadas para el gran influjo de datos que llega al servidor, y es más flexible, al no pedir los datos de una manera estructurada.

El servidor utiliza aplicaciones externas para ciertas tareas. Kepware se encarga de proporcionar los drivers necesarios para poder comunicar los autómatas con el servidor, y Wonderware permite almacenar los datos que llegan de los PLC, para así realizar un histórico que permita calcular el OEE (Overall Equipment Effectiveness) i identificar los KPI (Key Performance Indicators). Finalmente, el sistema MES permite visualizar todos los datos recogidos, y hace posible

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

establecer alarmas si los valores de las lecturas exceden las tolerancias predefinidas, para mantener la eficiencia de la producción.

Cómo utilizan la idea otros ámbitos

En ingeniería es complicado encontrar un concepto no aplicable a toda una serie de ámbitos. ¿Cómo se utilizan las tecnologías tratadas fuera de la industria?

Aunque el concepto de MES está pensado para aplicarse en entornos de producción industrial, la idea de automatizar grandes sistemas, de manera que se pueda obtener lecturas y operar actuadores en tiempo real, es aplicable a una inmensa cantidad de ámbitos. En la industria los SCADA se encargan de llevar a cabo este propósito, pero es interesante investigar qué soluciones se utilizan en otros ámbitos de la tecnología. Para ello es interesante saber qué necesidades existían, cómo se solucionaban en el pasado, y qué aproximación se toma en la actualidad teniendo en cuenta los medios que nos ofrecen los sistemas de automatización y control por ordenador.

En esencia, cuando se quiere controlar en tiempo real un proceso industrial, o parte de él, si simplificamos el proceso, lo que sucede es que se produce una interacción con la interfaz del sistema, y eso genera una señal que llega a los actuadores de manera que se pueda producir un fenómeno físico. Pueden existir eventos intermedios, y producirse acciones disparadas por la propia interacción, pero ese es el fundamento del concepto. Entonces, ¿por qué debería de abrir manualmente una válvula de un silo?, ¿por qué no podría tener una pantalla para controlar el rumbo de un transatlántico?

Uno de los ejemplos más flagrantes son los sistemas fly-by-wire que implementan los aviones modernos. A continuación, se muestra una comparativa entre los sistemas de control de los antiguos aeroplanos, y cómo se controlan hoy en día los aviones:



Ilustración 51: Controles de Vuelo

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

La imagen superior muestra los controles de un prototipo de avión utilizado en la primera guerra mundial. Se puede apreciar que, aunque consta de medidores, y sistemas de control, no hay indicios de ningún tipo de automatización ni tratamiento digital de datos. La palanca bajo el panel está conectada directamente al sistema de dirección de la máquina, y a excepción de algún sistema de engranajes que alivie el esfuerzo físico del piloto, no hay ninguna otra facilidad. Este tipo de aviones, además, contaban con motores que necesitaban un estímulo externo para iniciarse, por ejemplo, un operario que hiciese girar las hélices mientras el piloto manejaba los controles.



Ilustración 52: Fly-by-Wire

Esta otra imagen muestra los controles de un avión de pasajeros actual. Se puede apreciar cómo la mayoría de parámetros del avión se pueden controlar o se muestran de manera digital, mediante pantallas. También consta de controles tipo botón o palanca, pero no interactúan directamente contra los actuadores que accionan: la palanca central no maneja de manera directa los motores, y los controles del panel superior no son simples interruptores de paso de corriente. Son solo una interfaz para indicarle al ordenador central del avión con qué parámetros tienen que funcionar los sistemas. Eso en última instancia se traduce a mover los flaps que controlan la altura, o a hacer desplegarse los equipos de emergencia, entre otros muchos.

Otra prueba de los beneficios de automatizar algunos procesos son los sistemas ABS (Anti-lock Break System). Estos se basan en realizar pequeños frenados consecutivos, de manera que la reducción de velocidad se lleve a cabo más rápidamente y con menos pérdida de tracción.

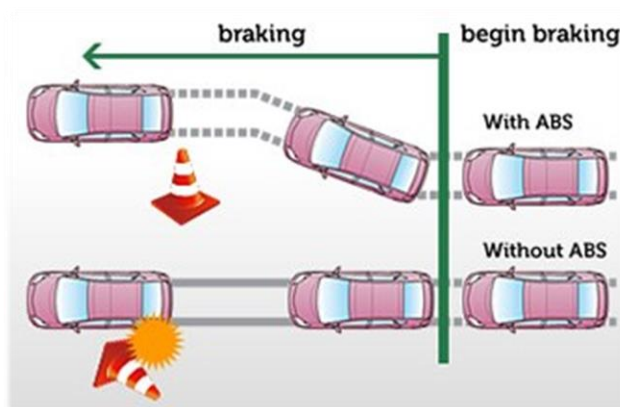


Ilustración 53: Beneficios ABS

Como se muestra en la imagen, el vehículo equipado con ABS consigue evitar el cono, mientras que el que carece de él, lo derriba. Esta capacidad de maniobra puede marcar la diferencia entre chocar contra un viandante o evitarlo, por ejemplo.

Este efecto se consigue instalando sensores de velocidad en las ruedas, de manera que cuando exceden una cierta velocidad y se produce un frenado brusco, se activa el módulo de control ABS, que activa un sistema de bombeo hidráulico de aceite para realizar los frenados.

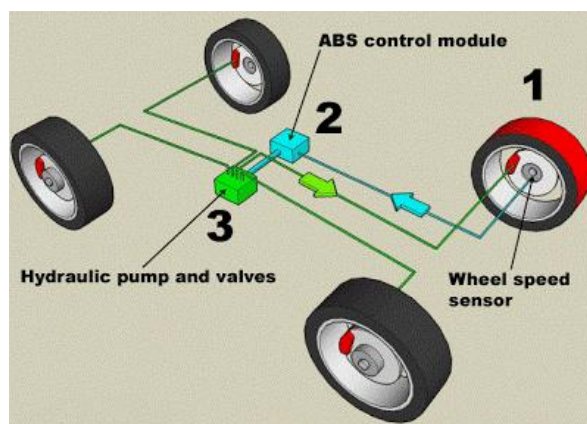


Ilustración 54: Mecanismo ABS

A continuación, se mencionan algunos ejemplos reales de compañías que han asimilado el concepto de industria 4.0 y han conseguido diferenciarse en su ámbito de negocio:

Logística

En los puertos de Rotterdam y Hamburgo, se ha rediseñado el proceso de transporte de contenedores, de manera que no se necesite intervención humana hasta el momento de transportar éstos con camiones. Se crea

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

entonces una zona de trabajo automatizado, al exterior de la cual los operarios siguen en trabajo que aún no se ha delegado a las máquinas.



Ilustración 55: Zona Libre de Personas

Esto es posible gracias a máquinas como los transportes o grúas sin contenedor. La filosofía que siguen estas instalaciones, y lo que contempla la ley es lo siguiente: si hay humanos involucrados, las máquinas deben ser controladas por humanos. O de manera inversa: si no hay humanos alrededor, las máquinas pueden trabajar de manera automática. Ésta es la máxima que siguen, y es lo que hace posible reducir la intervención humana a su mínima expresión.



Ilustración 56: Transporte Autónomo



Ilustración 57: Grúas Autónomas

Construcción

En los últimos 20-30 años, la eficiencia en el sector de la construcción se ha estancado, pero en la actualidad, el uso de herramientas más avanzadas que las usadas hasta ahora promete un gran avance en el sector. Uno de los ejemplos más flagrantes es el Semi-Automated Mason de Construction Robotics. Aunque requiere de un apoyo humano para completar el ciclo de trabajo, se encarga de la parte más laboriosa y lenta de este: colocar el cemento en los ladrillos y posicionar estos de manera adecuada. Mientras el brazo robótico construye la pared, el operario carga el depósito de ladrillos y cemento.



Ilustración 58: Semi-Automated Mason

Otro ejemplo es el sistema de conducción automática de Built Robotics, que permite automatizar el trabajo de algunas máquinas cuando el operador está ausente. De esta manera, el operario sigue teniendo su puesto de trabajo, mientras que parte de las tareas se pueden confiar a la máquina.



Ilustración 59: Built Robotics

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Transporte público

Si se pudiese calcular el espacio que necesitan los coches de una autovía para viajar a la máxima velocidad posible y no correr riesgo de chocar con el vehículo de enfrente, se podría disponer los vehículos de manera que se aprovechara al máximo el espacio y se optimizase el tiempo de viaje. En el este de Londres utilizan un concepto parecido para su red de trenes.



Ilustración 60: Bucle de Detección

Los vehículos cuentan con un sistema informático que detecta el cambio de polaridad producida cada vez que los vagones pasan por encima de cada bucle de cableado, y calcula la posición del convoy gracias a los parámetros leídos. De esta manera se puede gestionar en tiempo real la posición de los trenes, y tomar decisiones en tiempo real.

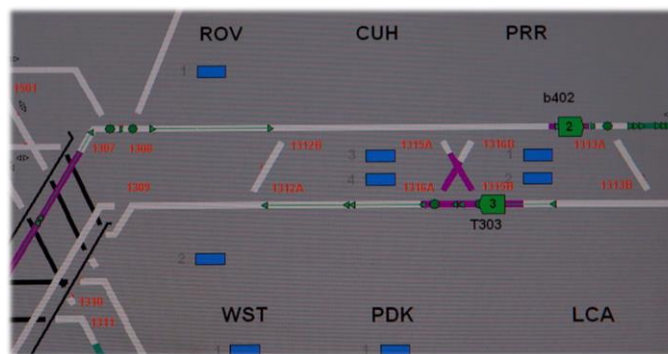




Ilustración 62: Plataformas Móviles

Un ejemplo de las tecnologías que utilizan son plataformas para transportar los chasis de los vehículos adheridos, o robots adheridos a estantes que permiten disponerlos en planta como se desee. Poseen también plataformas donde se instalan diversos brazos robóticos, que realizan tareas como las de atornillamiento o soldadura en la carrocería.



Ilustración 63: Estación Robótica Autónoma

Alimentación

Uno de los sectores más mecanizados es el de la alimentación, dado el alto grado de trabajo que conlleva. Desde ordeñar vacas, hasta realizar mezclas complejas de ingredientes y largos procesos de horneado, no se entiende la actual industria alimentaria sin toda una serie de máquinas que agilizan el proceso.

No importa si la tarea es tan “sencilla” como trocear una pieza de carne o elaborar una tarta: en la mayoría de casos, basta con que un operario posicione los ingredientes básicos en los depósitos de las máquinas y programar el proceso de fabricación del producto.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Comunicaciones: redes para la nueva industria

Una de las necesidades capitales de la nueva industria es la de comunicar al nivel más alto posible los dispositivos. Aquí veremos cómo se hace, y cómo se pretende hacer en el futuro.

Como en todos los ámbitos de la tecnología o la ingeniería, todo es un entramado de dependencias entre diferentes disciplinas, que hacen posible un resultado final orientado a satisfacer ciertas necesidades. La industria del futuro no es menos, puesto que se quiere integrar en el proceso de producción clásico toda una serie de dispositivos y funcionalidades, que necesitan comunicarse entre ellos, y con el exterior. Es aquí cuando entran en juego las redes industriales, que permitirán satisfacer esta premisa. Hay que tener un concepto claro: el éxito de la implantación de la industria 4.0 depende de un flujo de datos precisos y tratados en el tiempo adecuado, y esto a su vez depende de una red de comunicaciones conveniente, es decir que extraiga los datos de sus orígenes y los deposite en los softwares usados para su tratamiento. La imagen a continuación ejemplifica un esquema de red industrial, a nivel de corporación:

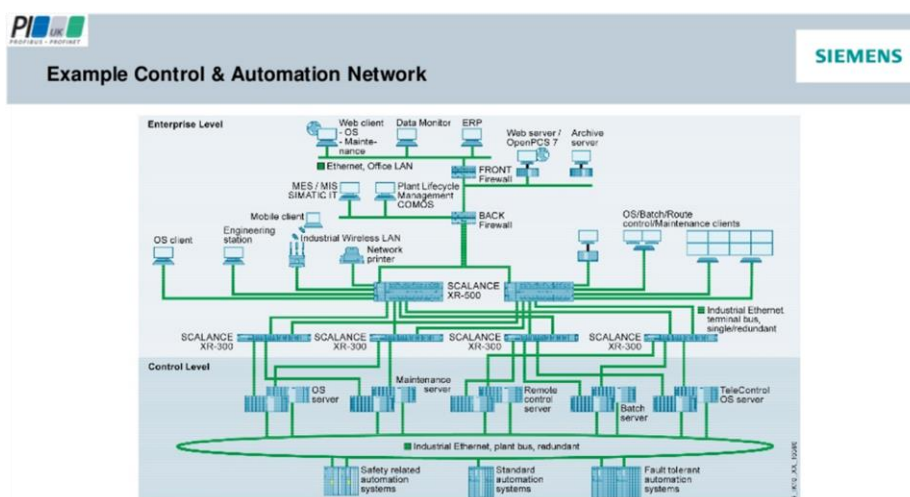


Ilustración 64: Ejemplo de Red Industrial

En su esencia, toda red se encarga de transmitir datos de un extremo a otro de las comunicaciones, pero hay que ser lo suficientemente sensible para apreciar las características de una instalación de comunicaciones industrial, y cómo contribuye a mejorar el nuevo paradigma industrial:

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES



Ilustración 65: Justificación de las Redes Industriales

El gráfico sobre este párrafo ilustra la idea desarrollada hasta ahora. Se parte de una red industrial que cumple los siguientes requisitos: asegura en gran medida la correcta y temporalmente conveniente entrega de datos del dispositivo servidor al dispositivo cliente (confiable), está diseñada para facilitar su adaptación tanto física como lógica a las necesidades de la instalación (escalable), y es lo suficientemente resistente a ataques y fallas de seguridad (segura). Con semejante medio de comunicación, se realiza una adquisición de datos que asegura precisión, haciendo posible un análisis de calidad de éstos que contribuya a un proceso de Mejora Continua productivo. Al final de la cadena está el éxito de la industria 4.0: una mejora en la eficiencia general de los procesos, de la calidad de los productos o servicios ofrecidos, y una subida de los beneficios como consecuencia de la potencial reducción de costes y la subida de la producción.

Entonces, si nos centramos en el diseño de la red, ¿qué principios se deberían de seguir? Previamente se ha discutido brevemente los detalles de una red industrial como dependencia de una adquisición de datos correcta, pero ahora pondremos el foco directamente sobre la red, remarcando las necesidades que ésta tiene que satisfacer:



Ilustración 66: Principios de las Redes

Interoperabilidad:

El objetivo es garantizar que se puede extraer datos de cualquier punto de la fábrica, y que los dispositivos que habitan en ella pueden comunicarse entre ellos. Actualmente existe la tendencia es la estandarización de Time Sensitive Networking, que provee de una ruta de comunicaciones común para las máquinas de planta.

TSN ofrece una única red Ethernet a través de toda la fábrica, permitiendo a los fabricantes un acceso rápido y eficiente de los datos. La tecnología está preparada para soportar aplicaciones punteras, como el control de robots, sistemas de visión o el Internet of Things industrial (IIoT), por lo que permite llevar el análisis y la realización de planes de mejora hasta todos los rincones del proceso productivo, eliminando posibles puntos que de usarse otros paradigmas de red serían inalcanzables. Además, el uso del estándar OPC-UA sobre TSN permite una comunicación segura desde el sensor hasta la nube, a la vez que ofrece protección para el tráfico crítico.

En tanto que, desplegado sobre Ethernet Industrial, TSN garantiza un flujo de paquetes conveniente, a la vez que asegura el bandwidth necesario y permite otras formas de tráfico sobre la misma red física, como por ejemplo la existencia simultánea de redes convencionales con redes en para sistemas en tiempo real. A continuación, se analizará TSN y se describirá qué puede aportar al ámbito que nos ocupa.

TSN como solución integral de tiempo real se fundamenta en el uso simultáneo de tres estándares del IEEE 802.1 que, en tanto que autosuficientes y de funcionalidad demostrada, contribuyen a la construcción de una tecnología mucho más completa:

Sincronización en el tiempo:

Todos los elementos participantes de la comunicación en tiempo real deben tener como referencia una base de tiempo común, de manera que tenga un contexto temporal común. El Ethernet convencional no es lo suficientemente riguroso en cuanto a tiempo como para garantizar la precisión requerida en sistemas con tanta intercomunicación y coordinación como los industriales. Para ello tanto las máquinas de planta como los equipos de red como los switches deben de tener un clock común, que les permita realizar ciertas acciones en momentos concretos. Para conseguir esta señal de sincronización común, se suele inyectar en la red una señal procedente de una fuente central de tiempo mediante el protocolo PTP (Precision Time Protocol), ya que el uso de GPS está sujeto a la cobertura de la planta, dificultando su uso en instalaciones móviles o subterráneas.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

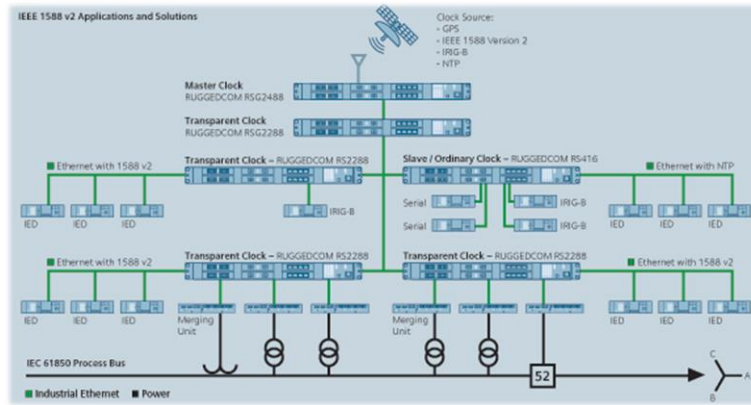


Ilustración 67: Sincronización Temporal

Traffic shaping y planificación:

Todos los elementos participantes de la comunicación en tiempo real deben seguir las mismas políticas de procesamiento y forwarding de paquetes. Para conseguir que los sistemas de tiempo real no sufran debido a la retención de sus paquetes en los buffers, se utiliza una modificación del scheduler IEEE 802.1Q con Time Division Multiple Access que elimina las interrupciones no deterministas del flujo de datos: el planificador IEEE 802.1Qbv.

Esta versión permite dividir el tiempo en ciclos, que se podrán subdividir en periodos más pequeños de manera que se puedan establecer varias políticas en lo tocante a qué paquetes viajan en cada fracción de tiempo. Como la siguiente imagen indica, esto permite localizar todos los paquetes con una VLAN priority de 3 en la primera franja temporal de cada ciclo, mientras que el resto de paquetes (con alguno de los 7 restantes niveles de prioridad que permite el estándar) viajan en el resto del tiempo. De esta manera, la VLAN priority no tiene que combatir con otros paquetes, mientras que cuando su tiempo finaliza, el resto se organizan según la política Strict Priority del protocolo.

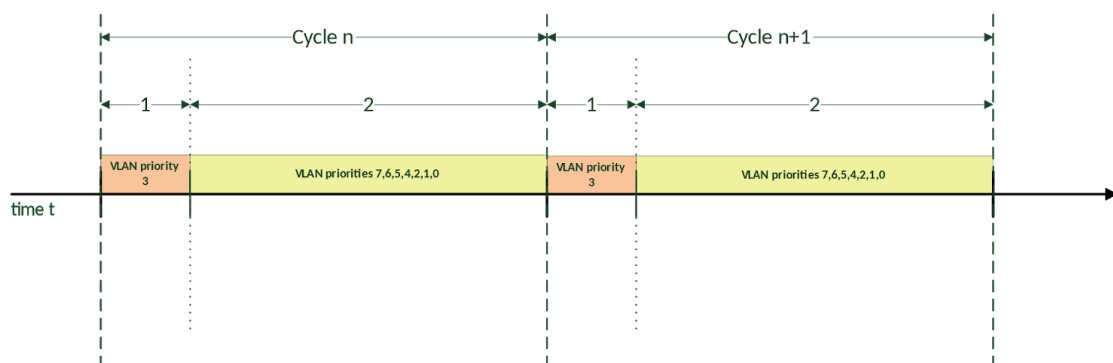


Ilustración 68: Franjas de Prioridades

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Pero esto es solo una aproximación a la solución final. ¿Qué pasaría si un paquete de una VLAN priority sin tiempo real (best effort) se solapa temporalmente con una que sí que tiene que cumplir los requisitos de tiempo real? La imagen a continuación ilustra el problema:

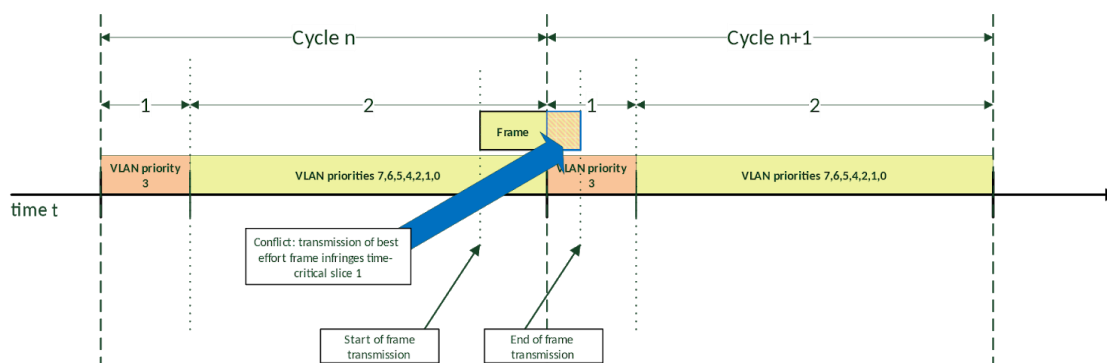


Ilustración 69: Solapamiento de Prioridades

Esto puede provocar que paquetes críticos sufran un retraso inadmisibles en algunas comunicaciones. Para evitarlo, se incluye un Guard Band frente a las franjas que transmiten paquetes críticos en tiempo. Éstas son tan largas como lo que se tarda en transmitir la medida máxima permitida de frame. El esquema entonces sería el siguiente:

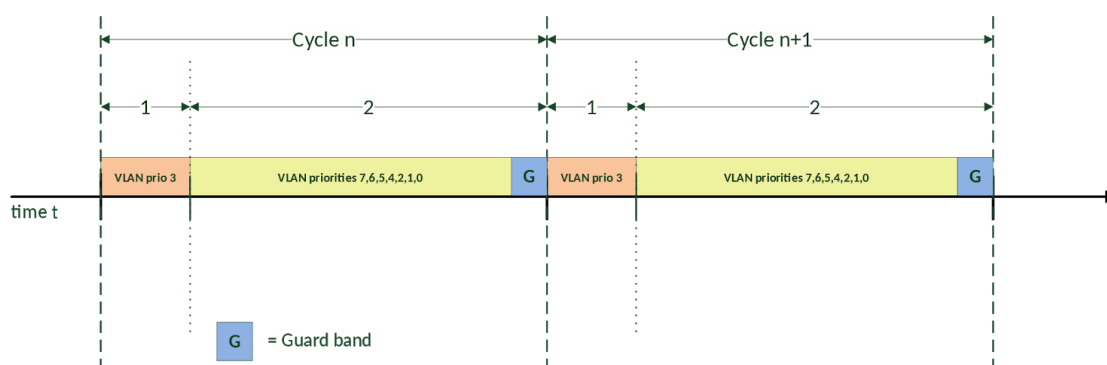


Ilustración 70: Guard Band

De esta manera queda asegurado que los paquetes críticos tienen un tratamiento adecuado respecto de los best effort. Lo siguiente para refinar el protocolo es mitigar los efectos negativos de la inclusión de esta Guard Band (incluye un tiempo en el cual el puerto Ethernet no puede transmitir, e introduce

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

penalizaciones considerables en sistemas lentos y de medida variable). Para ello se introduce una preempción en la transmisión de paquetes:

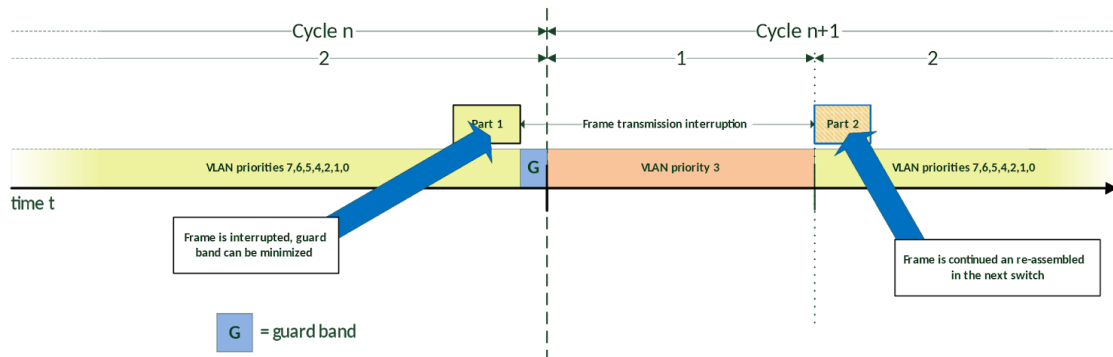


Ilustración 71: Preempción

Si el payload de un paquete no va a poder transmitirse dentro de su slice correspondiente, se separa y se añade un campo CRC al final del segmento, de manera que el payload restante se envía en el siguiente slice y el contenido del paquete llega en su integridad con mínimo impacto en los paquetes prioritarios. Esta aproximación reduce las penalizaciones a sistemas lentos, y posibilita también la reducción de tiempos de ciclo. La combinación de todos estos cambios tiene como resultado un estándar preparado para tratar paquetes de diversa naturaleza y satisfacer las necesidades de una red industrial que combina comunicaciones en tiempo real (robots), con comunicaciones best effort (chat interno, subida de datos a un histórico, etc.).

Selección de rutas de comunicación, reservado de rutas y tolerancia a fallos:

Todos los elementos participantes de la comunicación en tiempo real deben de seguir las mismas políticas de selección de rutas, reserva de bandwidth y slots temporales, y uso de varias rutas simultáneas para conseguir tolerancia a fallos. Actualmente se está desarrollando el estándar IEEE 802.1CB, pero hay otros dos protocolos altamente utilizados: HSR y PRP.

High-availability Seamless Redundancy confía en topologías tipo anillo para comunicar paquetes de manera redundante. Los nodos envían paquetes por más de un puerto de manera simultánea, hacen forwarding (a menos que sean el origen o destinación unicast del paquete), y los procesan si son el destino. Por razones de identificación, los paquetes incluyen una dirección de origen y un número de secuencia en la cabecera HSR. El uso de este sistema de redundancia podría sugerir que el bandwidth útil se reduce a la mitad, ya que cada paquete se transmite por duplicado, pero al doblar el número de cables, se consigue el ancho de banda nominal de la red.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

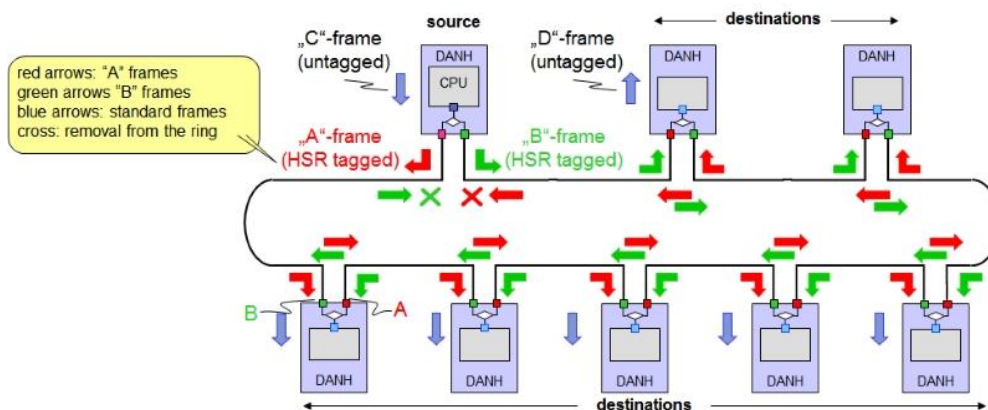


Ilustración 72: HSR

Parallel Redundancy Protocol, por otro lado, se estructura en nodos fuente y nodos destino. Cuando se envían frames duplicados, éstos viajan por sus respectivas VLAN, y al llegar a los nodos destino, estos se quedan con el primero de los frames, mientras que el segundo lo descartan. Como las dos LAN no se interconectan, las dos interfaces Ethernet de cada nodo pueden tener la misma dirección MAC sin producir conflicto. Al ser un mecanismo de redundancia a nivel 2, los protocolos de capas superiores no se ven afectados por su operación.

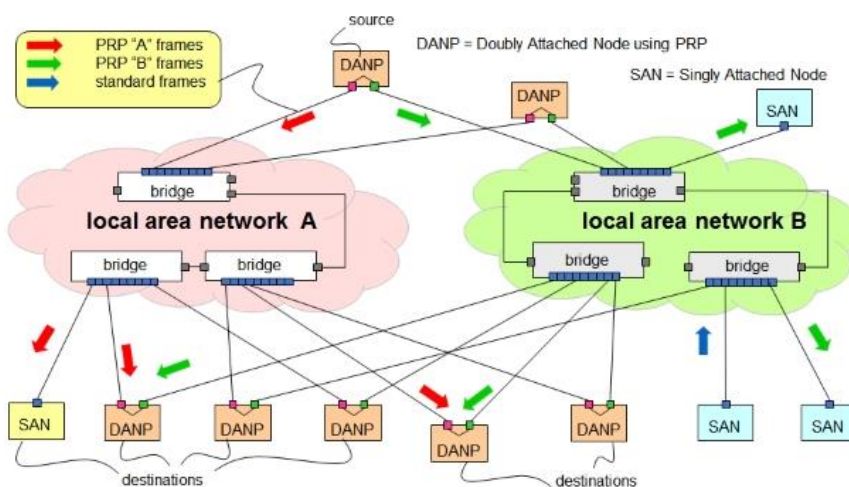


Ilustración 73: PRP

Automatización

En redes de una determinada complejidad, que crecen con la adaptación a la industria 4.0, no es plausible la configuración manual de los dispositivos. Las compañías pueden llegar a perder grandes cantidades de dinero derivadas de caídas de la red, por lo que, de caer, se necesita que se recupere lo más rápido posible. Para ello, resulta conveniente preparar el entorno para integrar equipos de manera 'plug and play', reduciendo al máximo los downtime, y haciendo

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

posible que éstos escojan automáticamente las opciones de configuración adecuadas.

Simplicidad

Para facilitar la gestión de la red, es importante escoger una infraestructura de red que sea capaz de manejar tanto TSN como Ethernet o Profinet, entre otros. Escoger el diseño adecuado de la red de comunicaciones hace posible que las máquinas puedan interactuar con el equipo de red (teniendo en cuenta los softwares de drivers intermedios).

Es importante también tener en cuenta que, aunque es el personal de Tecnologías de la Información quien se encarga de construir la red, también hay otro tipo de perfil que puede llegar a hacer un uso incluso más intensivo: los operarios. Es por esto que es importante utilizar soluciones que provean de herramientas intuitivas, de manera que, en caso de surgir algún error, se pueda reducir el tiempo para resolverlo. Si se confía solo en el personal de IT para solucionar estos problemas, se está desaprovechando una parte importante de la plantilla y se toma el riesgo de alargar en el tiempo un problema potencialmente simple de resolver, con las implicaciones que eso puede tener de cara al negocio. Algunas herramientas permiten identificar fácilmente puertos, redes y otros potenciales puntos de error, haciendo posible su identificación y corrección.

Algunas compañías, además, ofrecen diseños de red validados, que aseguran la correcta integración de sus equipos. Es buena idea usar este tipo de diseños, ya que contribuyen a facilitar la tarea de troubleshooting. Compañías como CISCO ofrecen diseños certificados adaptados según las necesidades de cada instalación. La siguiente captura de la web de CISCO ejemplifica las categorías que ofrecen:

Networking	Security	Data center
Campus wired and wireless LAN	All security guides	Big data
Data center networking	Cloud	Data center
Cisco Digital Network Architecture (Cisco DNA)	All cloud guides	Desktop Virtualization
Evolved programmable network	Industries	FlexPod platforms
Branch/WAN and Internet edge	Compliance	SAP
All networking guides	Manufacturing	All data center guides
Mobility	Oil and Gas	Collaboration
All mobility guides	Transportation	Remote Expert
Archive	Utilities	All collaboration guides
IPv6	All-industries guides	
Application performance		
IP video surveillance		
Medianet		

Ilustración 74: Categorías de Diseños

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Como se puede apreciar, existe una sección dedicada a Industria. Una muestra más de la importancia de las comunicaciones industriales. A continuación, utilizaremos la guía de CISCO para implementar redes en entornos de manufactura automatizada. Gracias a estas directrices (enfocadas al uso de equipos de dicha compañía), se puede tener una idea bastante precisa de la necesidad de este tipo de redes, de qué hacen para satisfacerla, y cómo lo hace. La siguiente imagen resume los objetivos de una empresa decidida a evolucionar hacia la automatización los desafíos que se encuentra y que ofrece una solución de automatización para resolverlos:



Ilustración 75: Objetivos y Propuestas



Ilustración 76: Barreras y Requerimientos

Por lo tanto, se quiere reducir riesgos, mientras que se aumenta la productividad general, se cumple con las regulaciones y se diferencia la compañía de la competencia, pero existen toda una serie de desafíos. La diferencia de conocimientos entre miembros de la plantilla, la dificultad de adaptar a los trabajadores con la edad, los problemas de seguridad derivados de la implantación de una red, el control y gestión de los datos recogidos y la falta de

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

estandarización del ámbito industrial en cuanto a hardware, software y protocolos.

Otro de los grandes problemas encontrados es el uso de redes desfasadas, que por razones históricas no se han renovado. Esto es muy común en la mayoría de compañías, que no ven necesaria la renovación de su equipo de red hasta que emprenden un proyecto ambicioso de automatización. Para paliar las limitaciones y cuellos de botella de las redes antiguas, una solución es optar por hacer converger todo el tráfico de la compañía en una salida al cloud. Como toda solución de Tecnologías de la Información que incluya el uso de la nube, esto tiene ventajas e inconvenientes. Al delegar procesamiento a la nube, hace falta menos hardware propio, y se ahorra en espacio, pero al confiar en una compañía externa, se crean posibles fuentes de error, ya sea en lo tocante a seguridad o, por ejemplo, por cuestiones de congestión en la red tanto interna como externa.

A grandes rasgos, la jerarquía en zonas de una red integral sería la siguiente:

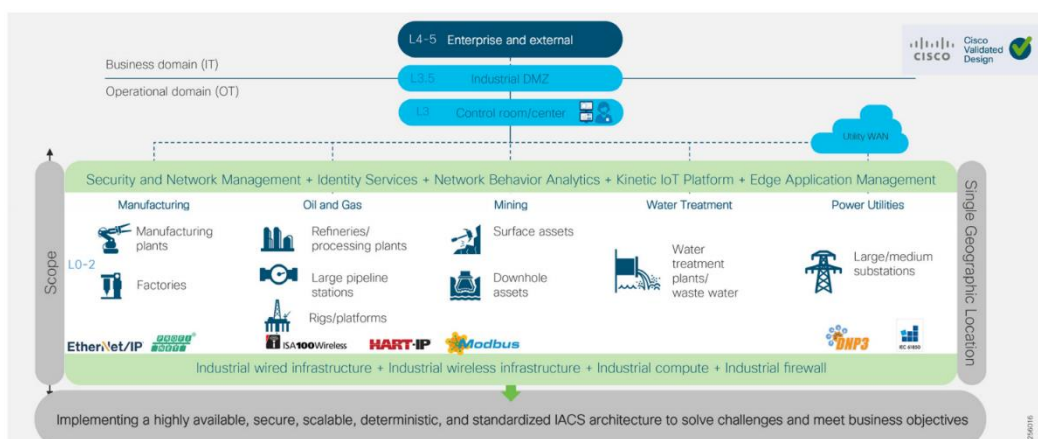


Ilustración 77: Jerarquización en Zonas

De manera más formal, las zonas se subdividen en los siguientes niveles:

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

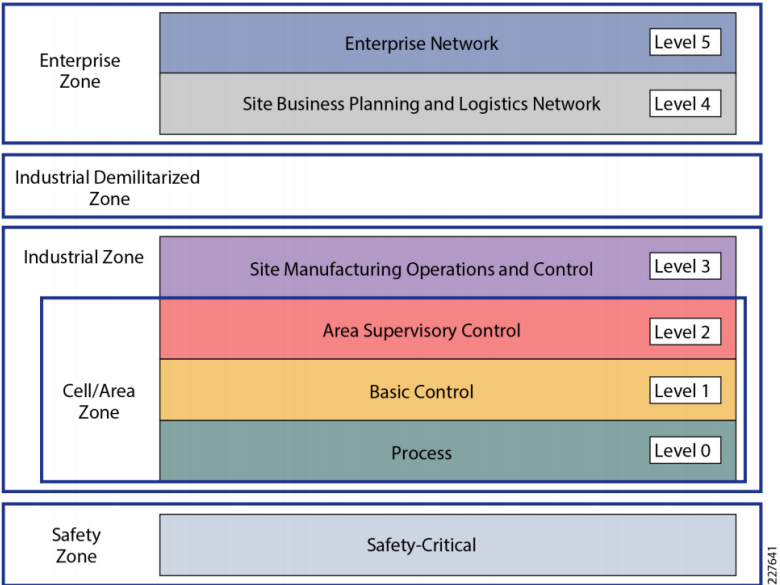


Ilustración 78: Niveles por Zona

Safety Zone

A nivel de planta, hay multitud de equipos que trabajan con voltajes altos, o que por razones mecánicas suponen un riesgo para los operarios. Sin las debidas medidas de seguridad, se podría manipular incorrectamente el equipo y comprometer tanto la integridad física de los trabajadores como el flujo de trabajo de la compañía. Para ello las redes dedicadas a seguridad están construidas sobre el IACS (Industrial Automation Control System), y aisladas, de manera que puedan cumplir con los estándares de seguridad aplicados a estas instalaciones, más restrictivos que los aplicados en otras instalaciones similares.

Cell/Area Zones: Acceso y Control

Esta es la región donde se lleva a cabo el grueso del proceso productivo. Aquí se distinguen zonas, donde se llevan a cabo partes específicas del proceso, y cells, o partes del proceso de una zona. Un ejemplo sería una fábrica automovilística: una zona podría encargarse del ensamblado del chasis, mientras que otra se encarga de la electrónica y la mecánica, y otra de la pintura los acabados finales. Una vez se tiene clara la lógica de subdivisiones del proceso productivo, se puede diseñar una red que se adecue a la estructura lógica de la empresa:

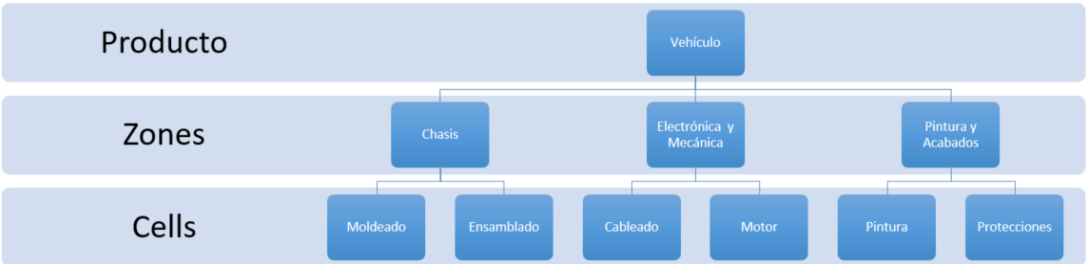


Ilustración 79: Zonas y Cells

El nivel 0 (Process) consta de los elementos encargados de llevar a cabo las tareas más fundamentales del proceso, como soldar, pintar, sensar o dar valor a una señal. Este nivel recibe órdenes del nivel 1, y le comunica el estado de las máquinas a los dispositivos de control de éste. Por lo tanto, este nivel principalmente se encarga de la comunicación determinística en tiempo real, manejar variables de entrada y salida y funcionar en ambientes adversos y topologías restrictivas.

Pasando al nivel 1 (Basic Control), tenemos los equipos de control que dirigen los dispositivos del nivel 0, y se comunican con los equipos y herramientas del nivel 2. Los controladores IACS son la inteligencia del sistema, y trabajan en conjunto con una red que les permite interactuar entre ellos y con los equipos de los que se encargan. Están configurados para tomar decisiones y comunicarlás a los actuadores de planta, para que se lleve a cabo la producción. En conclusión, el flujo de tráfico creado por los controladores es el siguiente:

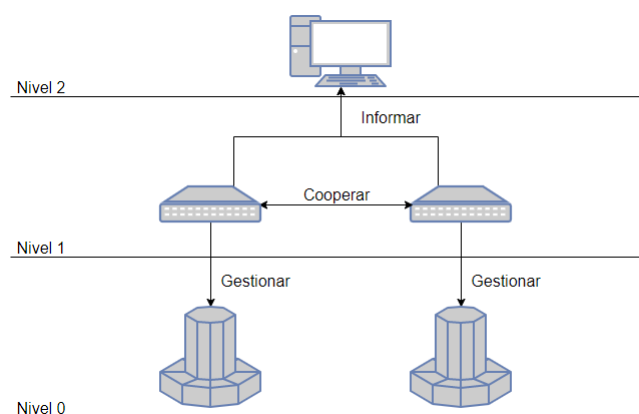


Ilustración 80: Flujo de Control

Por último, en esta zona, tenemos el nivel 2 (Area Supervisory Control). Aquí habitan todas las aplicaciones y equipos dedicados a las tareas de supervisión y operación. En este nivel los datos extraídos de los niveles inferiores se convierten información, y permitan tomar decisiones basados en ésta. Aquí también se establece la frontera a partir de la cual se comunican la zona de producción y la de supervisión. A grandes rasgos, encontramos HMIs para los operadores, sistemas de alarma y pantallas SCADA que permiten controlar las unidades de trabajo.

Industrial Zone

En el nivel 3 (Site Manufacturing Operations and Control) se incluyen todas las aplicaciones relacionadas con el control de la factoría. Entidades como un ERP o un MES quedan excluidas de éste nivel ya que están más enfocadas a cumplir funciones de negocio y supervisión. Softwares tipo SCADA residen a esta altura de la jerarquía, así como historizadores de datos y aplicaciones de control y

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

gestión de la red y la seguridad. Comúnmente los equipos aquí utilizados confían en sistemas operativos como Microsoft Windows o otros basados en Unix, por lo que muy probablemente los protocolos de red usados serán Ethernet y TCP/IP.

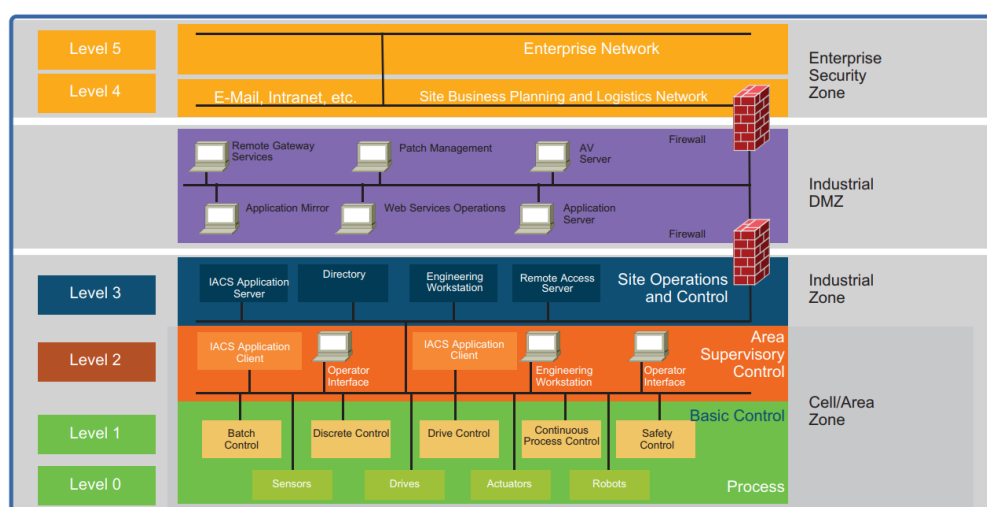
Enterprise Zone

En el nivel 4 (Site Business Planning and Logistics) residen todas las aplicaciones que necesitan de la red corporativa para funcionar. Pueden tener equipos conectados tanto por cable como de forma inalámbrica, y no suele intercambiar mucho tráfico con la red de producción. Esto es mayormente porque se considera la oficina como una potencial falla de seguridad de cara a planta, por lo que se diseña una zona desmilitarizada entre estas dos secciones de la compañía.

Es aquí donde un hipotético MES o ERP puede existir, así como otras aplicaciones cuyo propósito no sea influir de manera crítica en el proceso productivo (como sí es el objetivo de los niveles inferiores). Las funciones principales de éste nivel son, entre otras, permitir acceder a Internet, al servicio de correo, a aplicaciones como SAP, y realizar tareas de análisis y planificación con la información de las herramientas instaladas. La interacción entre este nivel y los más bajos es bastante reducida, produciéndose el mayor flujo de datos (en torno al 90%) hacia Internet o aplicaciones de cloud.

En el nivel 5 (Enterprise) residen los sistemas centralizados de Tecnologías de la Información, así como servicios de negocio a negocio y de negocio a cliente. Se suele aislar este nivel del IACS, haciendo que las aplicaciones corporativas hagan de interfaz para los sistemas de control, pero se pueden hacer excepciones, como conexiones inalámbricas para permitir que integradores de sistemas o fabricantes de equipo puedan acceder. Sin importar si el personal es interno o ajeno a la empresa, todo el tráfico que llegue a la red del IACS debe pasar por la zona desmilitarizada, de tal manera que se ejecuten los análisis y políticas diseñados por la empresa.

Demilitarized Zone



258205

Ilustración 81: DMZ y Elementos por Nivel

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Además de cumplir la típica función de las DMZ de alojar los servicios más vulnerables como el correo, DNS y los servidores de la empresa, en industria también hace las veces de aislamiento entre la red de planta y la de oficina, lo que aísla la primera de Internet. Para mantener los recursos críticos alejados de la influencia exterior, se inhabilitan las conexiones entre las dos partes principales de la red, aunque se mantiene el intercambio de datos para que los sistemas que necesitan intercomunicación sigan funcionando. El tiempo de inactividad que un sistema sufre después de un hipotético ataque resulta muy costoso para la compañía, y si la red de oficina es el principal vector de ataque, hace falta aislar lo máximo posible.

Hoy en día es extrañamente común que los ordenadores en planta utilicen sistemas operativos y aplicaciones desfasadas, que no están preparadas para resistir los ciberataques de la actualidad. Esta problemática abre la puerta a un gran número de potenciales amenazas. A continuación, se muestra en detalle la construcción de la DMZ industrial:

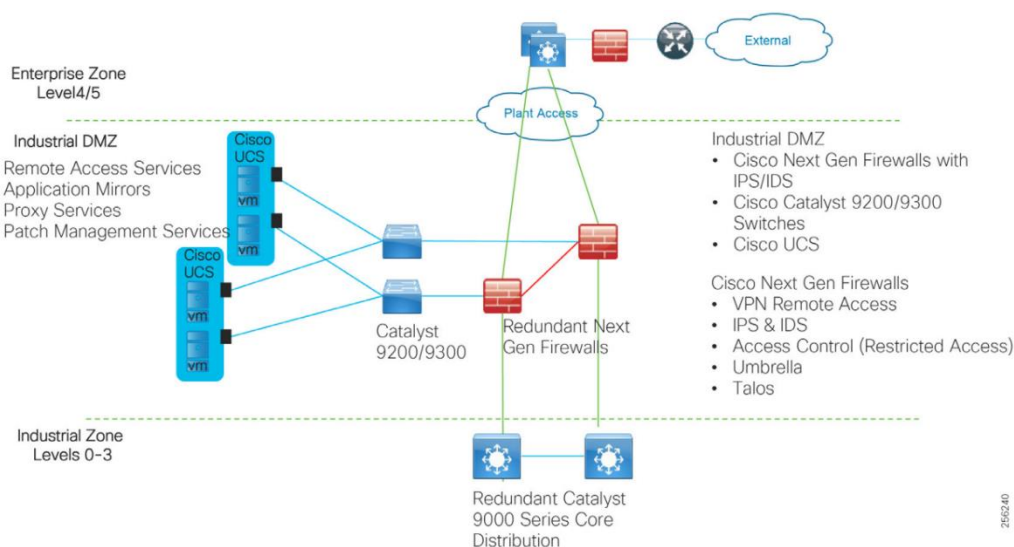


Ilustración 82: Diseño DMZ

De la imagen sobre estas líneas podemos extraer algunas conclusiones de cara al diseño de una IDMZ (Industrial DeMilitarized Zone):

- Como norma general, no debería de existir tráfico de datos entre la zona de oficinas y la zona industrial. Esto no es aplicable al uso de aplicaciones de oficina en la zona industrial.
- La IDMZ tiene que ofrecer comunicaciones seguras entre los extremos de la red mediante servidores replicados o en espejo.
- Se tiene que posibilitar el acceso remoto a los sistemas de la zona industrial.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

- Se tiene que diseñar una barrera de seguridad entre oficina y planta, aplicando las políticas adecuadas para restringir el tráfico al mínimo funcional.
- Las comprobaciones son bidireccionales: también hay que aplicar políticas para el tráfico que va desde planta hasta las oficinas.

Opciones de arquitectura

En el momento de diseñar el Core Network del sistema, hay que barajar varias posibilidades. La primera es mantener separado el Core del Distribution. De esta manera se goza de las prestaciones de un Core diferenciado en cuanto a redundancia, agregación y conectividad, y no se sacrifica escalabilidad. Esto hace el primer diseño adecuado para grandes corporaciones, y hace que el cambio de equipos no interrumpa el servicio.

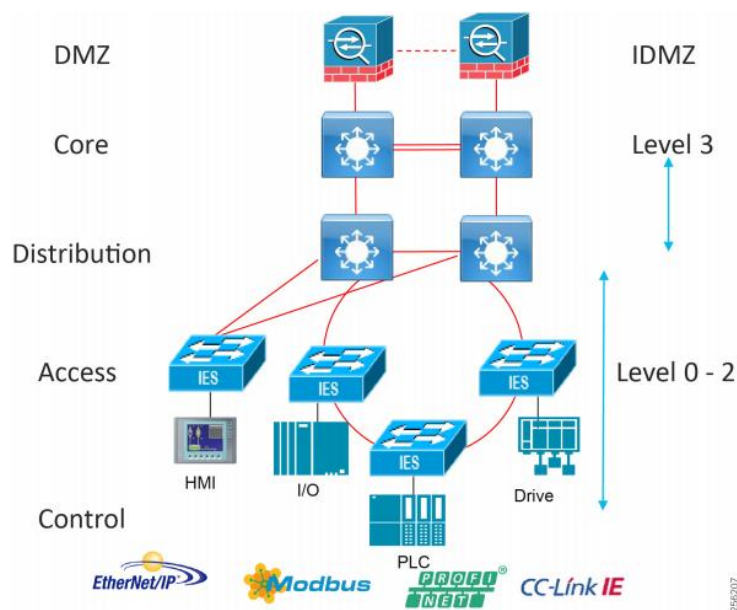


Ilustración 83: Separación Core-Distribution

Otra opción es colapsar el Core y el Distribution en una misma entidad, de manera que se simplifica la estructura de la red y se ahorra equipo, pero se pierde el aislamiento y agregación del diseño por separado, además de hacerlo susceptible a congestionarse. Este modelo no es tan escalable como el anterior, y está pensado para empresas de tamaño reducido a medio.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

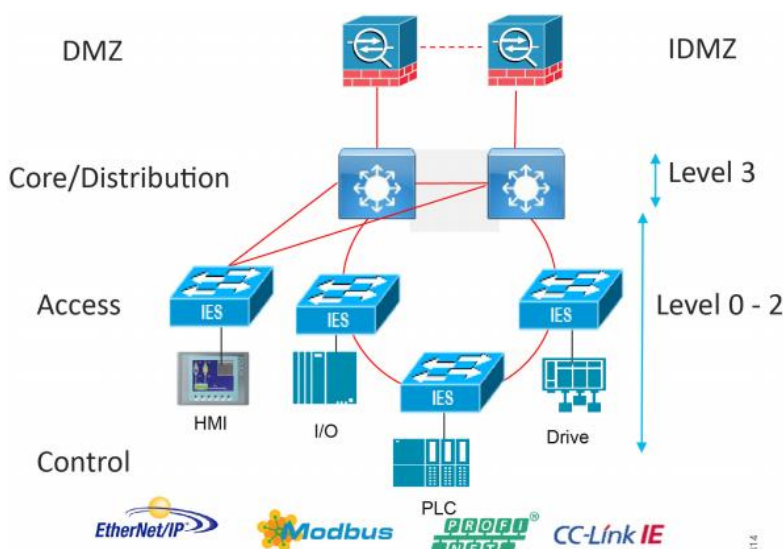


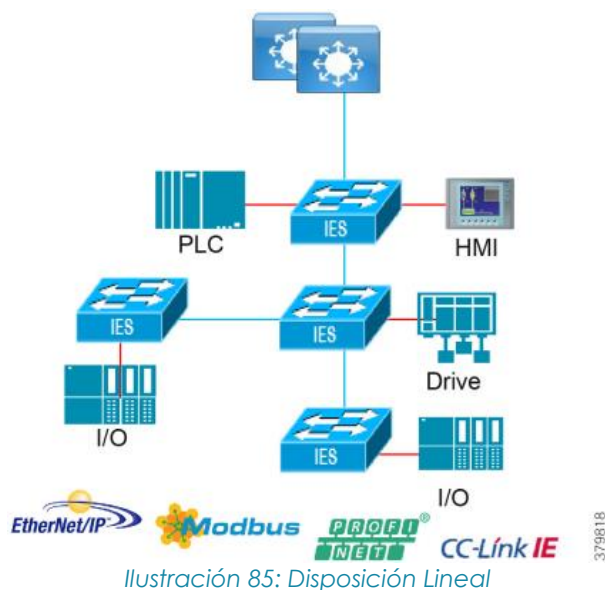
Ilustración 84: Core-Distribution Colapsados

A continuación, se describen varias configuraciones para el área de Cell/Zone, es decir, el Access de la red industrial. Esta parte de la red aloja el IACS, y hay algunos puntos clave a tener en cuenta a la hora de escoger topología:

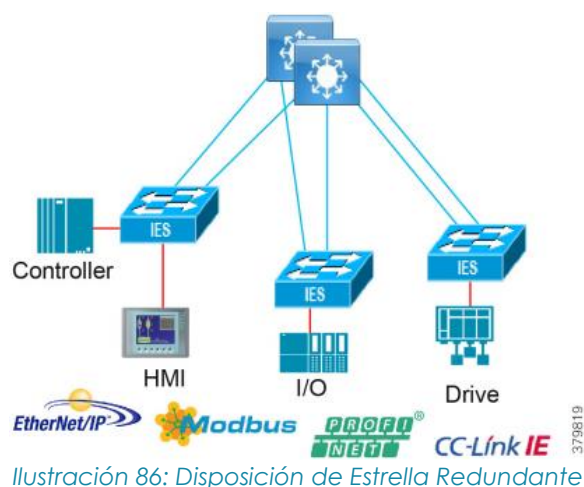
- Disposición física: Cómo esté dispuesta la zona donde se pretende desplegar la red influye en la instalación física de ésta. El cableado es mucho más caro en el entorno industrial, por lo que topologías en estrella pueden resultar imposibles por cuestiones de coste. Para tales casos, un diseño en anillo podría ser la solución.
- Disponibilidad: Éste es un parámetro clave para el cálculo del OEE, por lo que desplegar una red capaz de seguir funcionando después de la pérdida de un enlace o de hardware resulta muy conveniente. El tiempo de convergencia después de un cambio en la topología podría conllevar igualmente a un tiempo de inactividad, y a una caída temporal del IACS, pero un despliegue correcto contribuirá a reducir ese periodo.
- Comunicaciones en tiempo real: Las aplicaciones del IACS tienen que poder comunicarse a través de la red con un cierto grado de confiabilidad y predictibilidad. El tráfico tiene que poder fluir por saltos, cambios de latencia, jitter y otras fluctuaciones de la red manteniendo una Quality of Service adecuada.

La primera opción es disponer de los switches de manera lineal. Esto simplifica la instalación y reduce el cableado, pero abre la puerta a la aparición de cuellos de botella entre la Distribution y el switch con el que se comunica.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES



Otra opción es construir el acceso con una topología de estrella redundante. En esta disposición hay solo dos saltos entre todo dispositivo. Esto contribuye a la predictibilidad del tráfico, y el cableado doble contribuye a facilitar la convergencia de la red en caso de pérdida de enlace. También resulta más caro al tener que instalar más cables.



La tercera alternativa es una topología en anillo donde el último switch de un diseño lineal está conectado a Distribution, de manera que, si un enlace cae, siempre hay otro camino para llegar al destino. Esta arquitectura es más simple que una en estrella, reduce costes y es susceptible de usar algoritmos de redundancia como HSR o PRP.

Sumarizando todas las ideas vistas hasta ahora se obtiene el siguiente esquema, que sirve como guía para desarrollar un despliegue adecuado para una empresa que pretenda utilizar un IACS a nivel de planta.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

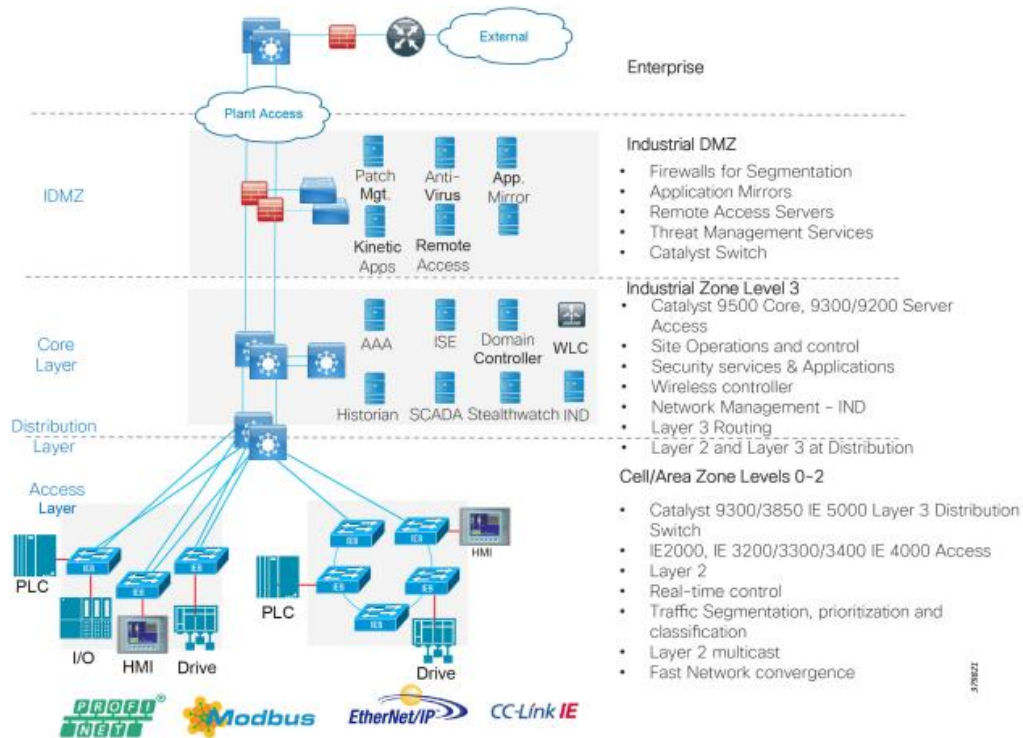


Ilustración 87: Esquema General

Seguridad

Con la creciente importancia de los subsistemas y los datos en la industria, la integridad y seguridad de éstos se convierten en elementos clave para las corporaciones. La ingente cantidad de dispositivos y datos hacen que una supervisión manual sea no sólo ineficaz, sino ineficiente, y descarta totalmente cualquier estrategia basada en “parchear” los problemas según surgen: se necesita seguir principios y guías como la ISO 21827 para conseguir algo lo más parecido posible a un sistema seguro. A principios de milenio, los ciberataques estaban enfocados mayoritariamente hacia las personas, por lo que los principales tipos de incidencias estaban relacionadas con el malware, los ataques de ingeniería social o el “defacing” de páginas web. Hoy en día, la tendencia apunta a una crecida de los ataques enfocados al crimen financiero, el espionaje industrial e incluso infraestructuras críticas y gubernamentales.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

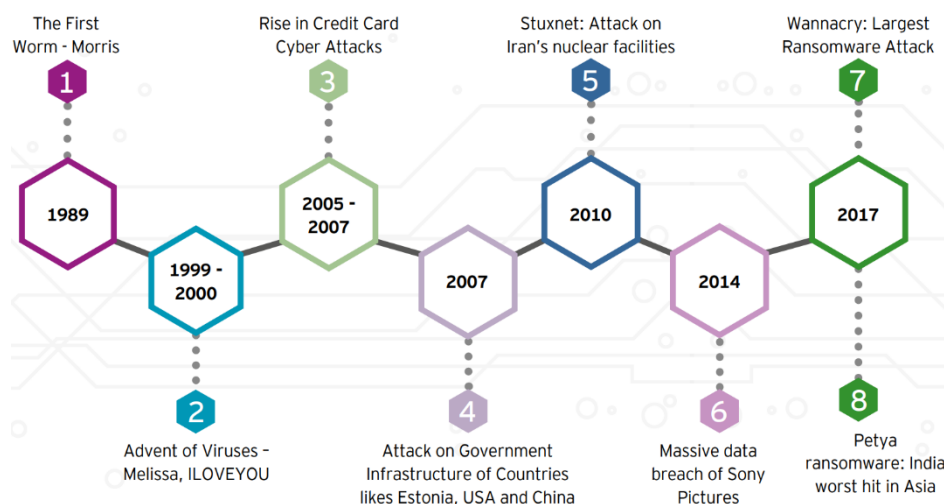


Ilustración 88: Evolución Ciberataques

Por lo tanto, se necesitan guías para asegurar la correcta aplicación de la ingeniería de la seguridad. Un ejemplo de modelo es el ya mencionado ISO 21287. Éste no establece reglas estrictas, pero captura algunas prácticas comúnmente adoptadas, cuya eficacia está probada. En líneas generales, el modelo cubre lo siguiente:

- Ciclo de vida completo del sistema: Desde el desarrollo al desmantelamiento, pasando por las tareas operativas y de mantenimiento.
- Organización integral: Actividades de gestión, organización e ingeniería.
- Interacciones concurrentes entre disciplinas: Sistemas, software, hardware, operaciones, mantenimiento, etc.
- Interacciones con otras organizaciones: Adquisición, gestión de sistemas, certificación, acreditación y evaluación.

Pese a poder aplicar un gran número de modelos, lo más importante es adoptar una filosofía transversal en cuanto a la seguridad de la organización. De esta manera, las medidas no se adoptan solamente en los equipos de red, ni las aplica el personal técnico, sino que empiezan en cada trabajador, en el día a día, y se infiltran en cada aspecto de la empresa. Es la mejor manera de aproximarse a un sistema perfectamente seguro, y de alinearse con los intereses de las partes que participan del negocio de la compañía.

Las tecnologías que participan de la Industria 4.0, y por lo tanto contribuyen a convertirse en fallas de seguridad, son muchas. A continuación, se mencionan algunas y se mencionan cómo sus potenciales vulnerabilidades pueden afectar a la actividad industrial.

Internet of Things

Son innegables los beneficios que una red de dispositivos interconectados aporta a cualquier proceso industrial. Desde el control de la plantilla hasta la interacción en tiempo real con los actuadores de producción, poder acceder a virtualmente cualquier lugar de la compañía y tomar decisiones con los datos recogidos es un valor añadido que marca la diferencia, y para hacerlo de

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

manera segura hay que conocer todos los actores que participan de esta nube de funcionalidades que es el IoT.

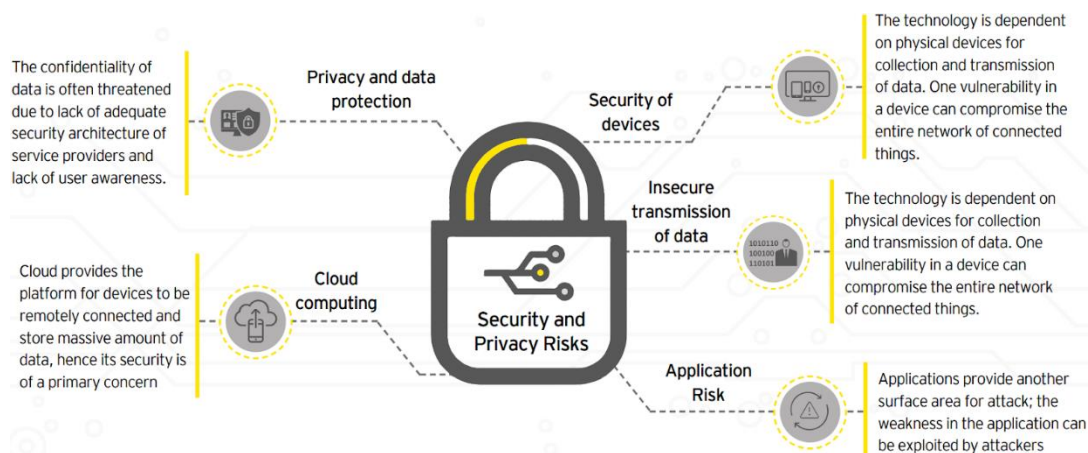


Ilustración 89: Riesgos IoT

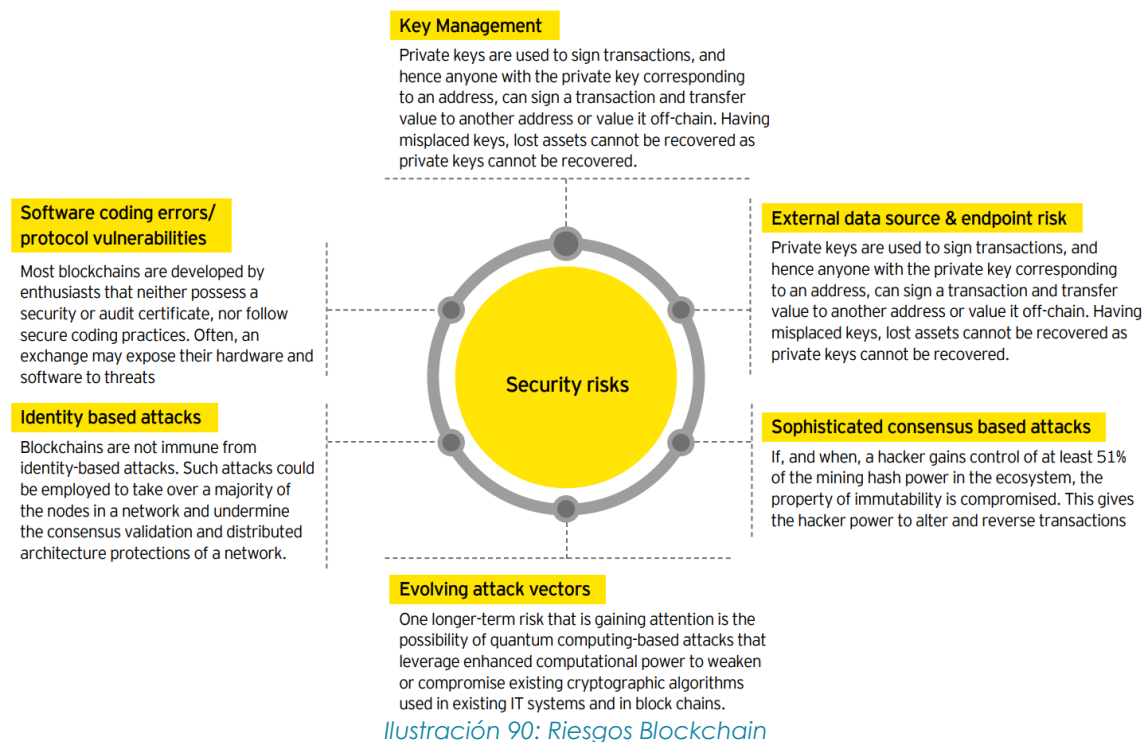
La arquitectura basada en servicios, por ejemplo, ve en su fundamento un problema de seguridad de gran importancia: el punto en la red que aloja el servicio y que, por lo tanto, provee de datos, puede a la vez exponer datos confidenciales si no se siguen reglas de arquitectura adecuadas y se educa a los usuarios a utilizarlos de manera sensata. De la misma manera el uso de la nube hace que también sea un punto crítico, por la dependencia que pueden llegar a tener las compañías en el almacenamiento y la capacidad de proceso que puede tener. Son solo algunos ejemplos de un uso descuidado de los recursos del IoT.

La naturaleza interconectada de los sistemas IoT hacen que una vulnerabilidad explotada en uno de los dispositivos de la red pueda llegar a conceder acceso a todo el sistema, con la gravedad que esto conlleva si se da en ámbitos como el sanitario o de infraestructuras. No es adecuado aproximarse al problema de la seguridad después de que los ataques se produzcan, sino que hay que construir dispositivos y desarrollar software teniendo en cuenta la seguridad desde su diseño hasta su despliegue.

Blockchain

Esta tecnología es cada vez más usada en criptografía y transacciones, ya que se centra en la integridad e inmutabilidad de éstas. El uso de blockchain permite acelerar el proceso de transacciones, reducir el fraude gracias a su lógica de comprobaciones y eliminar los bottlenecks, debido a su naturaleza descentralizada.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES



Aunque con mucho potencial, aún está en proceso de maduración, y por lo tanto abundan las malas prácticas a la hora de usarlo. Desarrolladores sin las acreditaciones necesarias pueden intentar usar blockchain en sus proyectos, exponiendo su hardware y software a posibles atacantes. También la cuestión de las claves privadas es importante, ya que con éstas se firman las transacciones y se pueden transferir valores de manera ilícita. La inmutabilidad puede ser comprometida si un atacante gana el control de al menos el 51% del sistema, y el poder de la computación cuántica pone en entredicho la seguridad del blockchain en un futuro.

Una industria automatizada podría confiar en blockchain para llevar a cabo toda una serie de tareas como el manejo de identidades, el envío de documentos y el certificado de ciertos datos. Sí se lleva a cabo de manera segura, puede llevar a un éxito sin precedentes en éste ámbito, pero no se han de olvidar los riesgos que asociados.

Inteligencia Artificial

Basándose en redes neurales, machine learning, deep learning y algoritmos de Natural Language Processing, la Inteligencia Artificial es capaz de resolver problemas complejos imitando el razonamiento humano. Después de entrenarse con grandes cantidades de datos, las máquinas se vuelven capaces de reconocer patrones y actuar de manera similar a como una persona lo haría.

La capacidad de detección de patrones de la Inteligencia Artificial hace que se puedan detectar y analizar nuevos vectores de amenazas y patrones de comportamiento. De la misma forma, puede ayudar a detectar malas prácticas en el código, y en la configuración de la red.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

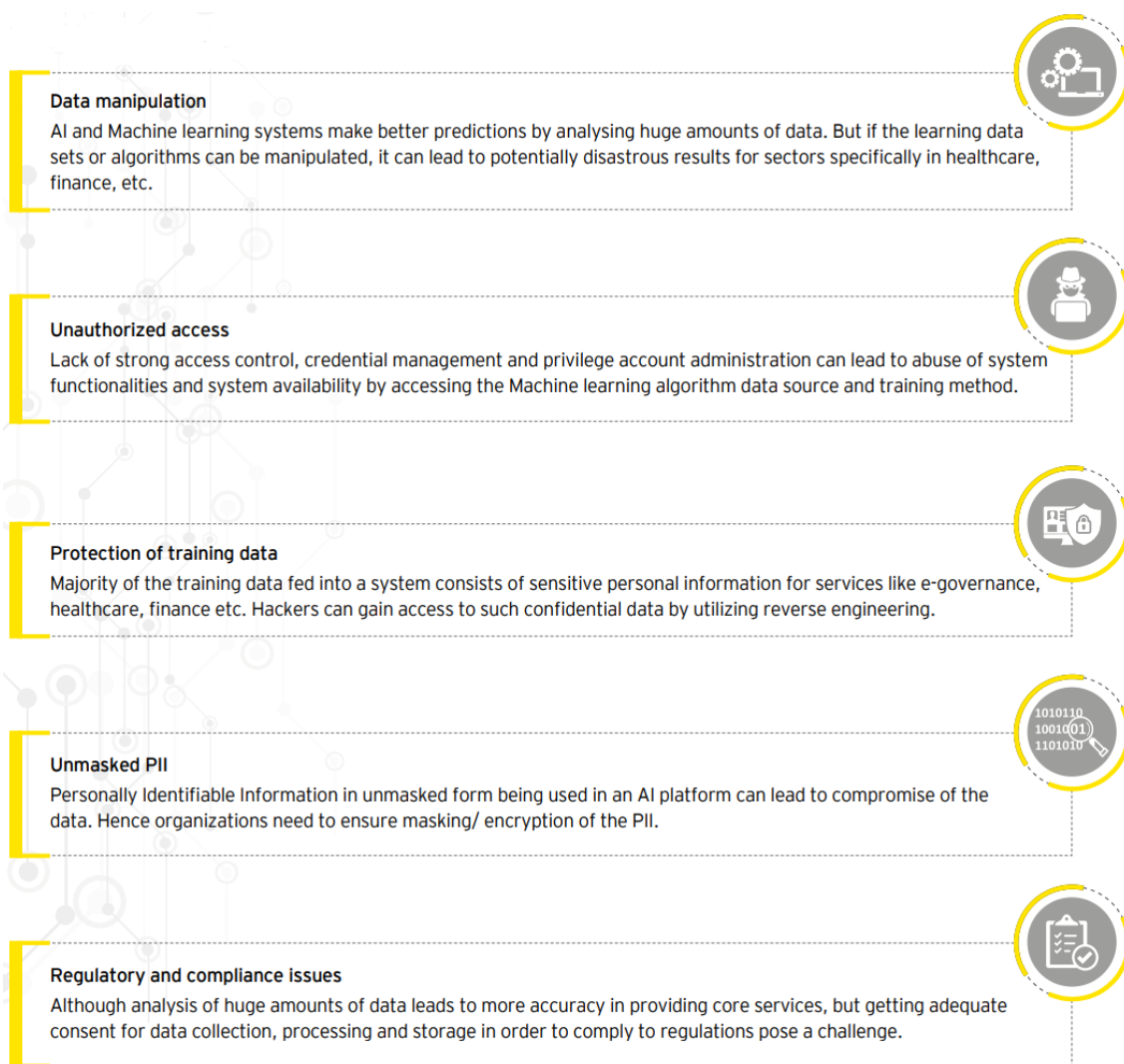


Ilustración 91: Vulnerabilidades Inteligencia Artificial

Como se puede observar en la imagen, hay varias maneras en las que un sistema de Inteligencia Artificial puede ser atacado, y por lo tanto el sistema industrial. Por ejemplo, una manipulación en la base de datos usada para entrenamiento puede llevar a la IA a tomar decisiones según los intereses de un atacante. De la misma manera, un acceso indebido a los algoritmos puede modificar el comportamiento de las máquinas. También es muy importante la protección de los datos en cuanto a su visibilidad: no se debe dejar a simple vista los datos utilizados, sobretudo en ámbitos como la sanidad, la economía o el gobierno, dada la sensibilidad de éstos.

Inteligencia

De la misma manera que una línea de producción funciona mejor si cada operario sabe cómo reaccionar ante cada situación sin tener que recurrir al encargado de planta, un sistema en red es más eficaz si los dispositivos que participan son capaces de tomar decisiones sin tener que delegar trabajo siempre al control centralizado.

Esta idea aplicada a las Tecnologías de la Información se conoce como el edge (o fog) computing. Consiste básicamente en procesar y almacenar datos lo más

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

cerca posible de dónde se necesita. Ésto lo aleja de los modelos centralizados centrados en el cloud, aunque es plausible un sistema de edge computing que confíe en parte en el poder de la nube. De hecho, parte de los beneficios de utilizar edge computing es la independencia que crea respecto al cloud, eliminando multitud de posibles vectores de ataque derivados del intercambio de tráfico de red hacia el exterior. La alta proximidad entre saltos de red en comparación con la distancia de comunicarse con un servidor externo hace que la latencia de los paquetes sea extremadamente baja, cumpliendo con los requerimientos de un sistema de tiempo real. Esta última característica puede incluso facilitar la comunicación de la planta con el ERP en las oficinas, eliminando el cloud de la ecuación para este propósito.

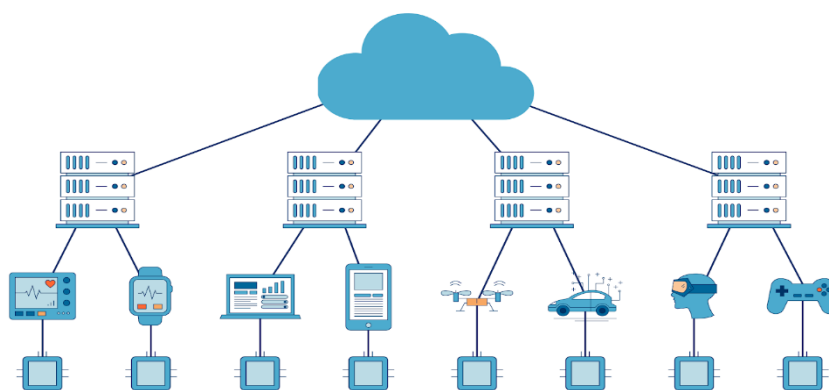


Ilustración 92: Edge Computing

Por modesto que sea el proyecto de automatización de una compañía, la naturaleza de la automatización hace que el volumen de datos crezca rápidamente. Actualmente existen herramientas y técnicas para Big Data que permitirían un filtrado personalizado de las lecturas, pero repartir el trabajo entre los dispositivos de la red contribuye a realizar un tratamiento previo que hace la recogida de datos hacia el sistema central más selectiva.

Economía y Sociedad

¿Cómo afecta a ésta revolución los factores sociales y de mercado?

En gran parte, el potencial de un territorio se mide según la fuerza de su sector industrial. No es de extrañar, por lo tanto, que grandes cambios en la industria tengan grandes implicaciones en la economía de la sociedad en la que se asienta, por eso es un aspecto con el que no se puede frivolizar.

Uno de los aspectos económicos más sensibles es la cuestión del empleo. ¿Desaparecerán puestos de trabajo? ¿Se crearán nuevos? ¿Cómo se reubicará a los empleados? Son algunas de las preguntas que más preocupan, y se han realizado varios estudios al respecto. La tendencia, al contrario de lo que podría parecer, es que se espera que se creen nuevos puestos de trabajo que sustituyan a los que desaparezcan, incluso que se creen más de los que se destruyen. Aproximadamente solo un 5% de los empleos serán completamente

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

automatizables, y se espera que las nuevas tecnologías generen empleos que cubran ese vacío.

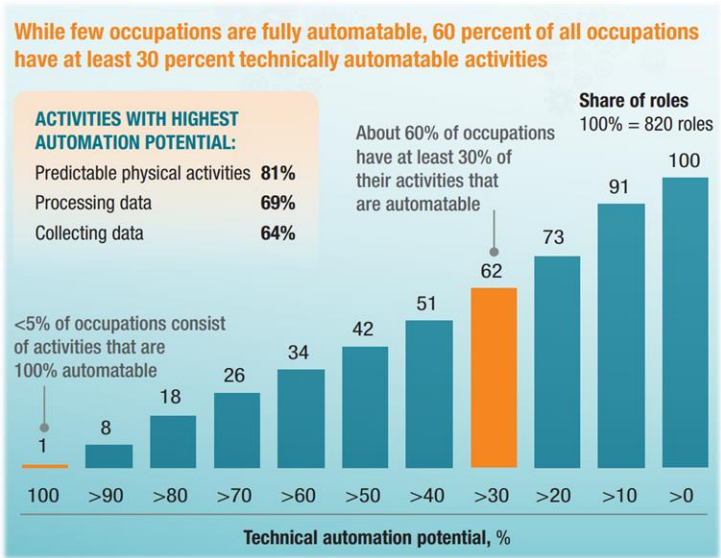


Ilustración 93: Susceptibilidad a Automatizarse

Los mismos estudios muestran grandes beneficios derivados de la correcta implantación de la Industria 4.0, pero para que se conviertan en realidad, hace falta que empleadores, empleados, y todo el ecosistema empresarial trabaje conjuntamente. Será necesario que humanos y máquinas trabajen en conjunto, pero para ello hay que invertir en la formación de la plantilla. El 60% de las ocupaciones tiene al menos un 30% de tareas que se pueden automatizar, lo que lleva a la siguiente conclusión: los empleados tendrán que trabajar cada vez más con la tecnología.

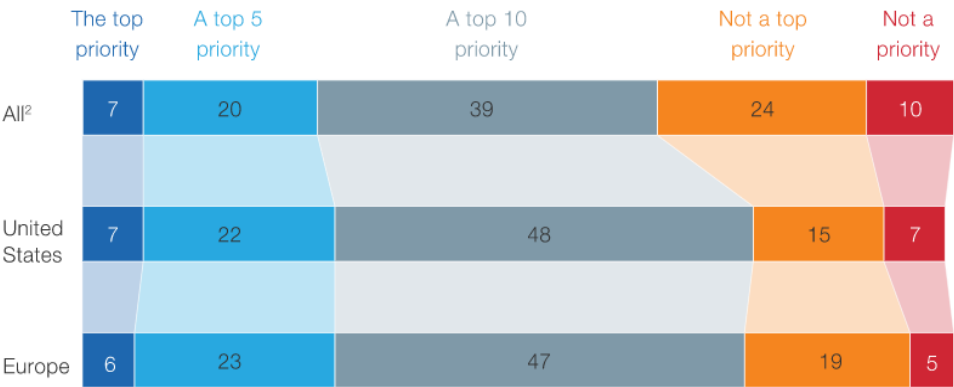


Ilustración 94: Sensibilidad a Evolucionar

En este aspecto, los trabajadores más cualificados saldrán beneficiados, ya que podrán aplicar más ampliamente sus conocimientos, mientras que los menos preparados notarán un aumento en su eficiencia y productividad. Cabe decir

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

que estos últimos, debido al desplazamiento del empleo, puede que sean muchos, y que por lo tanto noten cómo sus salarios se estancan o reducen, a menos que la oferta de empleo crezca lo suficiente para hacer caber gran parte de la demanda de empleo. El gráfico sobre este párrafo muestra, en porcentajes, cómo de importante se cree en el sector privado que es la formación de los empleados de cara a la cuarta revolución industrial. En promedio, un 20% piensan que es crítico, un 39% creen que es áltamente prioritario, y el resto no lo posicionan en posiciones de importancia o directamente no lo consideran. El siguiente gráfico ilustra el grado de automatización que admiten ciertos sectores, diferenciando los tipos de actividad que se llevan a cabo en cada uno y mostrando qué parte juegan en él (a mayor diámetro, mayor la importancia en tiempo), además de cómo de susceptibles son de automatizarse:

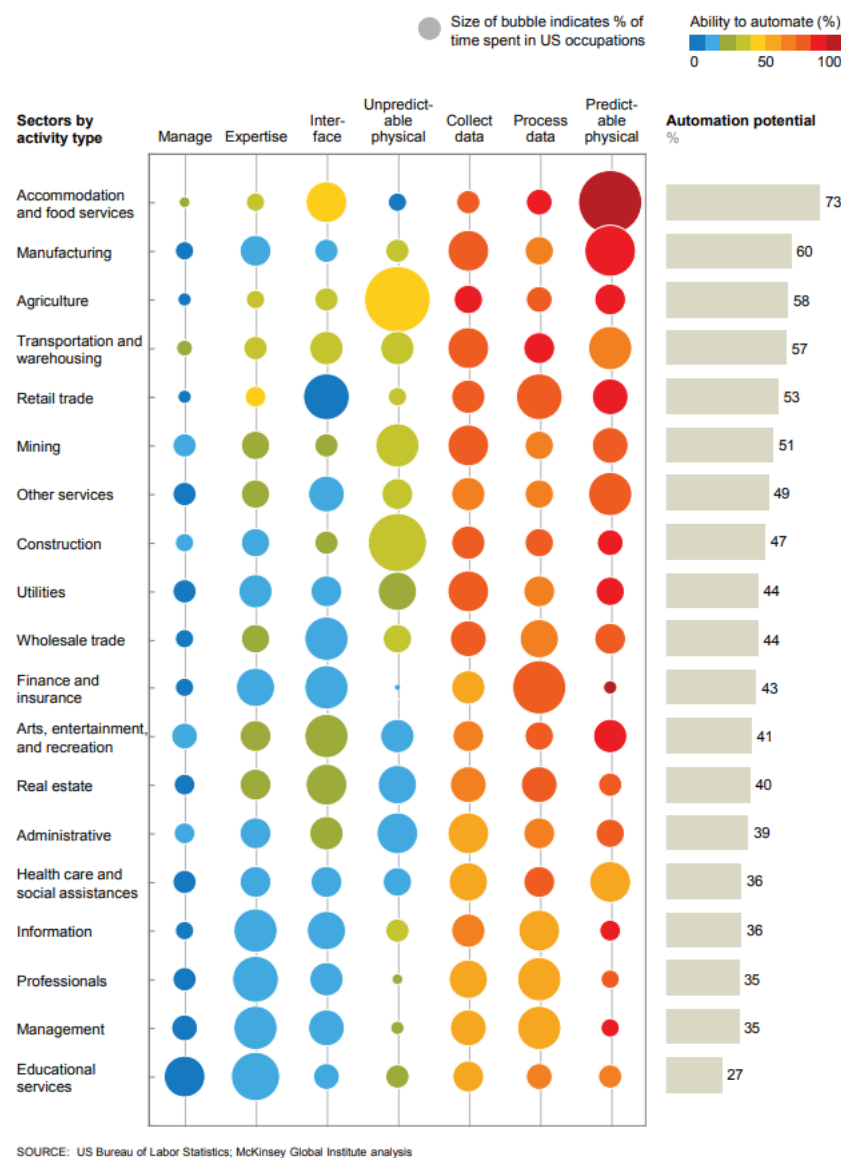


Ilustración 95: Previsión de Pérdida de Empleo Humano

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Según el gráfico, es previsible que en el futuro trabajos relacionados con la hostelería, como es el de camarero, se pierdan a favor de los robots. Los empleos consistentes en actividades físicas predecibles, así como el proceso y recogida de datos son los más susceptibles de ser automatizados, por lo que trabajos como el de camarero, albañil, o operario en una línea de montaje podrían verse desplazados eventualmente. Por otro lado, empleos que requieran grandes habilidades de gestión, experiencia, habilidades sociales o actividades físicas impredecibles tienen menos riesgo de desaparecer.

Pese a los evidentes beneficios en cuanto a negocio y productividad, la implantación de las tecnologías de automatización no depende solo de características técnicas, sino que está relacionada con toda una serie de aspectos transversales en la sociedad:



Ilustración 96: 5 Puntos de Reflexión

Plausibilidad técnica



Ilustración 97: Equipo Industrial

No se puede implantar una tecnología en estado de investigación. La industria tiene ámbitos altamente críticos, y sería irresponsable utilizarla como campo de

pruebas para nuevas tecnologías o ideas. Es por eso que hay que tener en cuenta el tiempo de desarrollo y maduración de las herramientas, así como la adaptación de éstas a cada entorno específico. Es el ejemplo de los dispositivos móviles como las tablets: no es lógico utilizar lo mismo para un entorno doméstico que para un entorno industrial, ya que las condiciones y necesidades no son las mismas.

Coste de desarrollar y desplegar soluciones

Para poder integrar a los procesos sistemas de automatización, es necesario desarrollar el software y el hardware sobre los cuales se sustentarán, y contabilizar el proceso de integración, con todos los contratiempos que eso puede suponer.

Dinámica del mercado laboral

Como se ha comentado anteriormente, la transición a un nuevo paradigma de producción conlleva consecuencias más allá de lo técnico. Las compañías llevan a cabo sus actividades gracias a sus empleados, y es responsabilidad de todo el panorama laboral preparar a la gente para poder rendir en la industria del futuro. Semejante revolución implicará desplazamientos de puestos de trabajo, por eso hay que velar por crear nuevos puestos de trabajo, y reubicar a la población activa de manera que el desempleo no se convierta en un problema de grandes dimensiones.

Beneficios económicos

Si a una compañía se le presenta claro que sustituir su mano de obra humana por robots representa una inversión rápidamente amortizable, probablemente la realicen. Este punto va en relación con el anterior, dado que habría que ver, en tal caso, hasta qué punto se es capaz de prescindir de empleados, o qué se decide hacer con empleados desplazados por máquinas. Aquí se presenta uno de los diversos conflictos morales que plantean grandes cambios de este tipo: ¿a cuánta gente se es capaz de dejar sin trabajo? ¿realmente las instituciones están preparadas para reubicar a los desempleados?

Aceptación social y regulación

Los prejuicios y costumbres sociales pueden ralentizar un proceso técnicamente beneficioso. La cultura popular puede moldear en gran medida la opinión de la gente acerca de la entrada de la tecnología en sus vidas, y forzar algo en contra de la voluntad de las masas puede tener consecuencias graves de cauce político y social.

Aunque está demostrado que las tecnologías actuales tendrían impacto en el 50% de la economía global, no es solo el aspecto técnico de ésta revolución lo que hay que tener en cuenta. El lugar dónde se pretende hacer la implantación tiene mucha importancia, ya sea por la oferta de trabajadores lo suficientemente preparados, o por la fuerza de la economía presente. Es posible que en algunos países exista la suficiente mano de obra como para que la implantación de grandes sistemas de automatización no aporte grandes beneficios. Por todas estas razones, no se espera que el nivel de automatización en los empleos llegue a niveles cercanos al 50% hasta dentro de dos décadas.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Un ejemplo real de protestas contra la pérdida de puestos de trabajo son las manifestaciones que realizaron los trabajadores del puerto de Rotterdam, a causa de la creciente automatización de su lógica de trabajo.



Ilustración 98: Manifestaciones en Rotterdam

A raíz de estas protestas, el gobierno accedió a no prescindir de trabajadores, e intentar reubicarlos en nuevos puestos de trabajo derivados de la aparición de las nuevas tecnologías. Aunque las instituciones y empresas fomentan un proceso de aprendizaje perpetuo, de manera que los trabajadores se puedan adaptar a los cambios del negocio, habría que preguntarse si es realista pretender que personas que no han querido formarse en sus primeros años estarán dispuestas a estudiar y renovarse continuamente.

Uno de los grandes argumentos que esgrimen los líderes de este tipo de protestas se puede resumir en el siguiente gráfico:



Ilustración 99: Cadena de Consecuencias

Es innegable el beneficio empresarial de la automatización, de la misma manera que es innegable que una máquina trabaja, en general, de manera más precisa y rápida que un humano. Lo que no hay que olvidar es que la gente sigue existiendo, y sigue teniendo necesidades, por lo que si pierden su empleo, pierden ingresos. Esto abre la siguiente cuestión: ¿quién pagará impuestos? Es decir, ¿quién pagará las escuelas, infraestructuras, hospitales y demás servicios que el estado sustenta en los impuestos recaudados? Una idea es aplicar impuestos especiales a las compañías con cierto grado de automatización, o ofrecer ayudas económicas a los desempleados por el mismo motivo. Como se puede apreciar, es una cuestión sensible sin una solución ni simple ni rápida, por

lo que lo más inteligente es planificar a años vista qué medidas se van a tomar al respecto.

Gestión Energética Corporativa (Corporate Energy Management)

Los CEM son las partes del sistema de automatización encargados del consumo de recursos. Aquí se justificarán tanto por sí mismos como dentro de este proyecto.

¿Por qué informatizar la gestión de consumos?

¿Por qué debería una compañía confiar en un sistema de gestión energética? ¿Resulta una inversión rentable?

Ya se ha mencionado la importancia que tienen los sistemas de automatización en el proceso de evolución tecnológica del sector industrial, pero ¿cuál es el sentido concreto del CEM dentro de estos sistemas?

Los procesos industriales y las infraestructuras que lo hacen posible carecen de transparencia en cuanto a su consumo energético, además de tener una naturaleza heterogénea. Es por eso que se necesita un sistema con todo un conjunto de interfaces, que haga posible eliminar esa falta de transparencia, y poder saber realmente cómo está funcionando el equipo, y lo que es incluso más importante, cómo está llevando a cabo su tarea.

Pasando a su aplicación más formal o administrativa, como ya se ejemplificó en la explicación del concepto de MES, no es realista pensar en equipos enteros de operarios dedicados exclusivamente a adquirir los datos de las máquinas y hacerlos constar en los archivos de la compañía, y menos de manera que esos datos sean fieles a la realidad. Esta necesidad de adquisición de datos es satisfecha en su integridad por los CEM, dado que brindan información en tiempo real recogida durante la producción, y la archivan, de manera que se pueda tanto detectar problemas "in situ", como realizar análisis posteriores para proyectos de mejora, o presentar pruebas con vigencia legal que hagan cumplir con la legislación a la que se someta la empresa.

Una vez se tiene clara la necesidad de los CEM, hace falta aclarar qué papel juegan estas herramientas en el proceso de cumplir el objetivo de toda empresa: obtener beneficios. El perfil de empresa que necesita un sistema automatizado de gestión de consumos es el de una compañía con diversos lugares encargados de la fabricación material de sus productos, y una dinámica económica donde, en el balance entre ingresos por venta y pérdidas, la inversión en recursos energéticos sean parte importante de esta última. El siguiente gráfico ilustra cómo un CEM puede contribuir a la crecida de beneficio de una compañía, vía la reducción de gastos en energía:

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

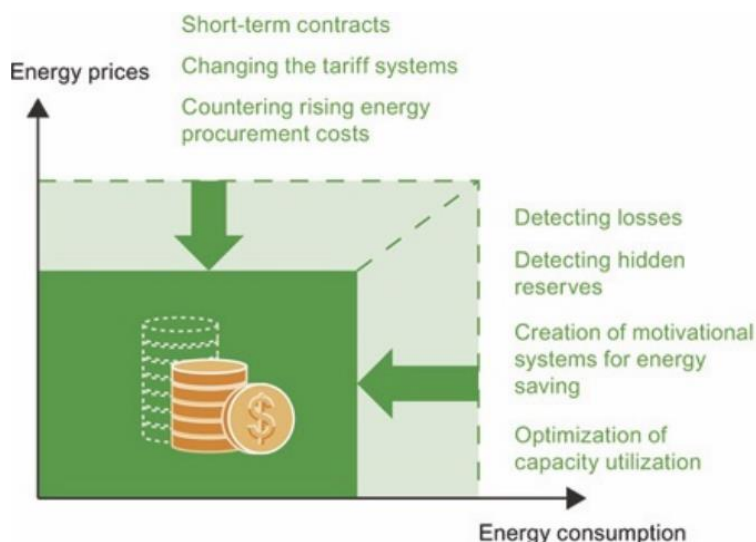


Ilustración 100: Efectos CEM

Por lo tanto, en cuanto al capital económico invertido en energía, las capacidades analíticas y predictivas de estos sistemas permiten revisar las tarifas vigentes, y replantearse los contratos con los proveedores, de manera que se pueda llegar a acuerdos más convenientes para la empresa. En cuanto al consumo en sí de energía, la detección de pérdidas o fugas en los procesos productivos, el descubrimiento de reservas inutilizadas, y la planificación de arquitecturas más eficientes harán que el consumo de recursos del proceso productivo se ajuste más al ideal.

Como apunte final, la tendencia actual es que las empresas, ya sea por publicidad, principios o conveniencia, están adoptando posiciones “verdes”, o lo que es lo mismo, están tomando medidas para que su actividad tenga el mínimo impacto posible en el medio ambiente, e incluso contribuya a su mejora en lo posible. Es por eso que el uso reconocido y eficaz de un CEM puede traer a la compañía beneficios tanto materiales, en forma de un aumento de los beneficios, como abstractos (buena publicidad, ayudas gubernamentales, etc.).

EnMPRO: Arquitectura y Escalabilidad

Estudio de la documentación oficial, con el objetivo de ser lo más fieles posible al propósito original de la herramienta.

Para que EnMPRO pueda cumplir su propósito, necesita toda una infraestructura que le brinde los datos que necesita para llevar a cabo su función. En rasgos generales, la arquitectura es la siguiente:

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

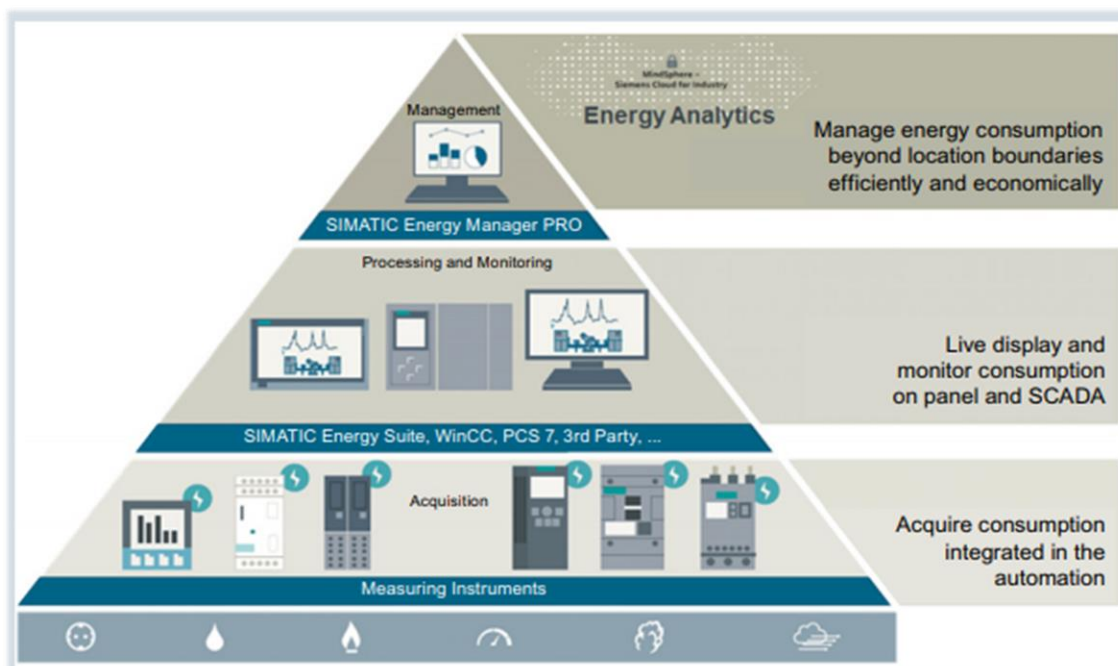


Ilustración 101: Pirámide EnMPRO

Como se puede apreciar en la pirámide, se distinguen tres niveles en la estructura completa. A continuación, se describirá cada uno, y se aclarará cómo contribuye al propósito general del CEM.

El nivel más bajo que encontramos es el de campo. Aquí se evidencia la filosofía principal de estos sistemas: empezar el ahorro energético desde lo más abajo posible en la lógica del negocio. Para ello se utilizan herramientas de alta precisión que brinden datos precisos, ya sean de consumo eléctrico como de otros tipos (gas, agua, etc.).

Subiendo un escalón nos encontramos con el nivel operativo y de control. Aquí se implementan todas las interfaces necesarias para comunicarse con los distintos equipos. Entre las tecnologías con las que el software trabaja se encuentran las siguientes: PROFIBUS, PROFINET, Modbus, M-Bus, S0-Impulse. Es aquí donde se implementa el punto de acceso único con el que EnMPRO trabaja.

Al final de la pirámide se encuentra el nivel de gestión. Es aquí donde habita EnMPRO, y donde se realizan los análisis y predicciones, la programación de planes de consumo, y la visualización de los datos y la información gracias a clientes desplegables en un navegador web, ya sea en ordenadores o en smartphones.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

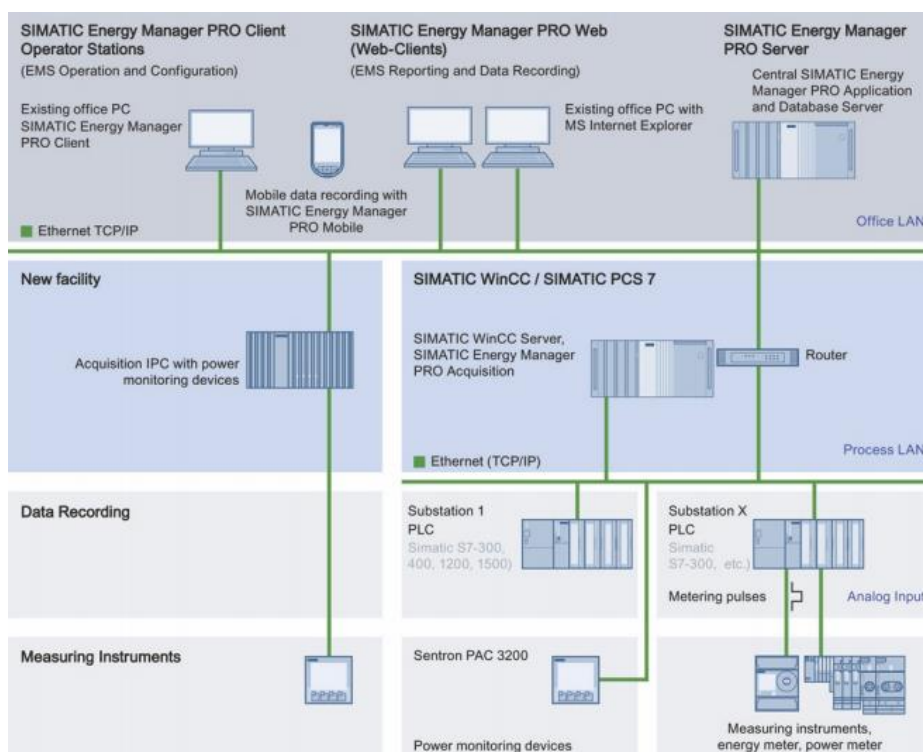


Ilustración 102: Arquitectura EnMPRO

Como se muestra en la imagen anterior, en el nivel más bajo, los equipos de medición comunican con los autómatas en planta, para luego enviar la información al servidor Adquisición de SIMATIC. Allí se crea el punto de acceso que el servidor de EnMPRO utilizará en el nivel de oficina. Ese mismo nivel es compartido por el sistema de gestión y los potenciales clientes del sistema. Según este paradigma, si se está conectado a la misma LAN en la que está instalada la infraestructura del CEM, con los softwares adecuados, se pueden desplegar clientes de manera prácticamente inmediata, y siempre de acuerdo con las licencias adquiridas junto al software. Los softwares de tunneling actuales permiten crear VPNs que harían posible el acceso remoto al sistema de gestión de consumos desde cualquier lugar. La siguiente imagen simplifica el esquema anterior:

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

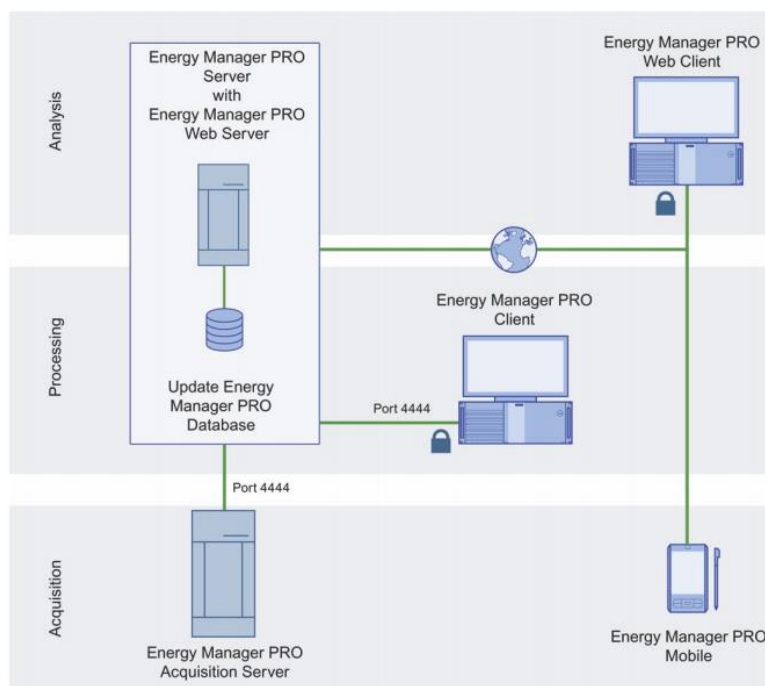


Ilustración 103: Jerarquía EnMPRO

En nivel de adquisición de datos, EnMPRO despliega un servidor que almacena las lecturas de las máquinas, para que se pueda acceder a ellas en el nivel de procesado, y analizarse posteriormente, en el nivel de análisis, a través del web client.

Es interesante también describir la infraestructura que existe debajo del nivel más inferior de toda la arquitectura explicada, es decir, los niveles lógicos que conforman el proceso desde que los sensores y actuadores interactúan con el mundo físico, hasta que los operadores y otros softwares pueden visualizar y manipular los datos que se les brinda. En este tipo de sistemas, se trabaja con una base de datos centralizada, que maneja todas las alarmas, señales y eventos del sistema, y una única herramienta de ingeniería, a través de la cual se pueden gestionar los controladores de la infraestructura y diseñar los objetos gráficos del sistema, además de servir de punto de entrada para propagar órdenes por todo el sistema. Son los sistemas conocidos como Distributed Control System. El siguiente gráfico ilustra la propuesta de este concepto, y servirá para ver cómo encaja “debajo” de la infraestructura que propone EnMPRO en su documentación.



Ilustración 104: Niveles del Sistema CEM

A nivel de elementos de campo, se busca integrar el instrumental con los actuadores de las máquinas, vía el bus de campo escogido en cada despliegue concreto, por ejemplo, Profibus o Fieldbus. Más arriba en la jerarquía se encuentra la periferia descentralizada, encargada de reducir al máximo la distancia entre los dispositivos de control y las máquinas de producción.

El concepto de periferia descentralizada surgió como solución a los problemas que plantea intentar adquirir de manera eficaz las señales de las máquinas, cuando existe una infraestructura compleja de cableado que cubre una distancia relativamente grande. Cablear de manera centralizada una red semejante comporta tener que lidiar con latencias elevadas y distancias no compatibles con algunos protocolos. Para solucionar esta casuística, se centralizan las señales de E/S a nivel de máquina mediante la instalación de un módulo capaz de agregar todas estas señales, y interpretar las órdenes que llegan del PLC que controla el equipo en cuestión. De esta manera se reduce el cableado a medida que se sube en la arquitectura, y permite una interconexión conveniente de los sistemas de control mediante una red industrial. Además, ha demostrado ser un modelo más escalable, cómodo en cuanto a cantidad de conexiones en los armarios, y conveniente de cara a la detección de fuentes de error. El siguiente gráfico muestra de manera simple esta idea.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

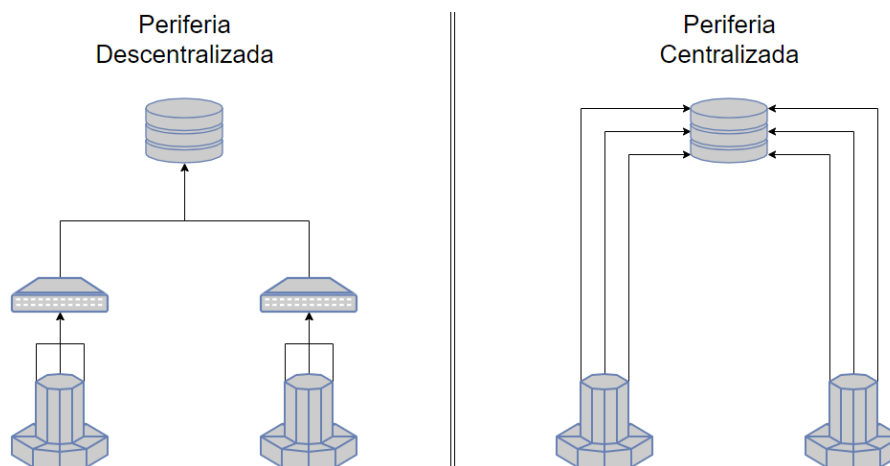


Ilustración 105: Diferentes Periferias

Visualmente es fácil apreciar los problemas de escalabilidad que plantea la Periferia Centralizada. Es cierto que la configuración es más sencilla, y requiere menos equipo, pero a menos que se trate de un sistema estanco, sin previsión de crecimiento, y con un cableado simple, este modelo no resulta práctico en la mayoría de casos.

Siguiendo con la arquitectura de los DCS, está el nivel de Control. Aquí es donde se distribuyen los equipos de control, es decir, los PLCs, y donde se diseña la lógica de comunicación entre ellos. Por último, en el nivel de Operación se produce la historización de los datos y la visualización en tiempo real de estos. La salida de los datos recogidos de los niveles inferiores servirá tanto para realizar planes de Mejora Continua como para alimentar otro tipo de sistemas, como un MES, un SCADA, o un CEM que utilice un EnMPRO.

En tanto que soluciones desplegadas en ambientes de producción, se exige que la disponibilidad sea virtualmente permanente, para minimizar al máximo posible el impacto que una caída del sistema de automatización provocaría en la producción planificada. Es por esto que los sistemas basados en DCS incluyen en su gran mayoría sistemas de redundancia, como PLCs, redes o buses adicionales, configurados para pasar a formar parte activa del sistema si su análogo cae por algún motivo.

Entorno de simulación

Descripción de la arquitectura de simulación utilizada.

El objetivo final de este proyecto, como se ha mencionado, es servir de punto de partida genérico para construir sistemas capaces de monitorizar los consumos de recursos en procesos industriales, y desarrollar una aplicación propia que pueda servir como complemento que haga todo más accesible. Pese a eso, no es realista pensar que las pruebas de funcionamiento y correcciones del sistema se vayan a hacer en un entorno de producción. En tal caso, los datos tratados corresponderían con la lógica real del negocio, pero es altamente probable que sea necesario sintonizar parámetros y corregir errores, antes de desplegar efectivamente el sistema.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Es por eso que se ha usado un entorno simulado, de manera que el flujo de datos ocurre de manera lo más similar posible a como ocurriría en realidad, pero sin poner en riesgo la actividad productiva de un hipotético usuario del sistema. El gráfico siguiente ilustra un despliegue real simplificado del sistema, sin concretar el software específico utilizado:

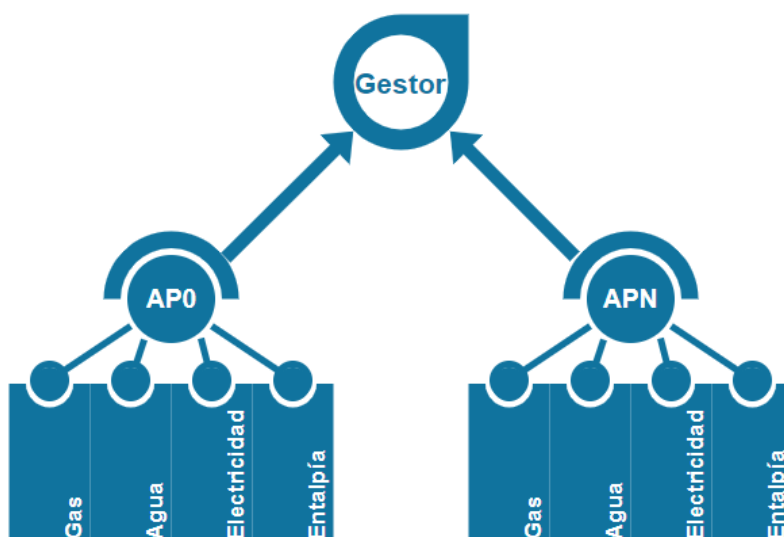


Ilustración 106: Entorno General de Simulación

En el nivel más bajo, se tiene toda la infraestructura de captación de señales, esto es, los sensores situados en los lugares adecuados, las redes y pasarelas como medio de transmisión físico, y los contadores, que cuantifican efectivamente el consumo.

Un paso más arriba en la jerarquía se encuentran los sistemas de adquisición (analizadores de planta APx). La función de éstos es la de recoger los datos de la infraestructura de captación, y dar un punto de acceso a los operarios, para que éstos puedan monitorizar y controlar, a nivel técnico y de planta, el consumo de los procesos industriales.

En último lugar, encabezando la jerarquía, se encuentra el gestor. Es la pieza de software encargada de sumarizar todo el flujo de datos provenientes de las diferentes plantas (es decir, de los diferentes analizadores). Ofrece, entonces, información a nivel corporativo y con enfoque empresarial, de cara a que sea el personal ejecutivo (gerencia) quien maneje este software. Para ello, debe de permitir generar gráficos y reportes, así como facturas basadas en el consumo medido, y realizar cálculos analíticos con el objetivo de confeccionar planes de mejora.

Una vez se tiene claro cuál es la estructura del sistema real, se puede abstraer en dos bloques, de manera que la simulación queda simplificada. Para mantenerse alejado de los pormenores de cada uno de los elementos de captación, se usa un autómata programado de tal manera que genera señales variables, de manera parecida a cómo lo harían los contadores en un

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

despliegue real. En conclusión, para parametrizar el software de gestión se tiene la siguiente arquitectura:



Ilustración 107: Arquitectura de la Simulación

Como se ha mencionado, el autómata alimenta el software de manera similar a cómo lo harán los contadores en el despliegue real, permitiendo establecer una parametrización altamente fiel a la necesaria en un hipotético entorno de producción. La comunicación autómata-gestor se realiza mediante IP, ya que el software se ha instalado en una máquina virtual que corre en un servidor instalado en la red, de manera similar a cómo se dispondrá al desplegarse el sistema. A continuación, se muestra una imagen del modelo de autómata utilizado, producido por Siemens:



Ilustración 108: Autómata

Y una imagen del montaje realizado para alimentar el autómata y conectarlo a la red:



Ilustración 109: Conexión del Autómata a la Red

Se puede apreciar cómo se conectó a la red mediante Ethernet, y cómo se preparó el cable de corriente para adaptarlo a las entradas disponibles en el dispositivo.

Es una buena idea confeccionar el trabajo de simulación de manera que se parezca lo máximo posible a un despliegue real, pero es mucho mejor realizarlo con herramientas que permitan avanzar todo el trabajo posible en tiempo de desarrollo, de manera que, en el momento de desplegar la solución, solo se tengan que cambiar algunos parámetros, como las direcciones IP. Para ello se creó una máquina virtual con Windows Server 2012, de manera que para hacer funcionar la solución en las máquinas de una compañía que adquiriera el sistema, solo se tendría que exportar la máquina virtual, instalarla en el servidor y cambiar los parámetros que sea necesario.

KEPServerEX: Simulación con PLC real

Gracias a mi estancia en una empresa del sector industrial, se realizó una simulación con equipos equivalentes a los usados en producción.

Con el PLC instalado en la red de las oficinas, es posible configurar contadores en un PLC, de manera que retorne valores similares a los recogidos por un sistema de captación real. Para ello, primero hemos de cerciorarnos de que el PLC es accesible, intentando contactar con el mediante un ping.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

```
PS C:\Users\amanco> ping 192.168.71.140

Haciendo ping a 192.168.71.140 con 32 bytes de datos:
Respuesta desde 192.168.71.140: bytes=32 tiempo=1ms TTL=30
Respuesta desde 192.168.71.140: bytes=32 tiempo=2ms TTL=30
Respuesta desde 192.168.71.140: bytes=32 tiempo=2ms TTL=30
Respuesta desde 192.168.71.140: bytes=32 tiempo=1ms TTL=30

Estadísticas de ping para 192.168.71.140:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
              (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 1ms, Máximo = 2ms, Media = 1ms
```

Ilustración 110: Conectividad con el PLC

Una vez estamos seguros de que el autómatas es accesible, programamos contadores a diferentes frecuencias, creando un segmento de programación por contador. Configuramos un pulso positivo, de manera que el incremento solo se produzca cuando la frecuencia configurada dicte, y no se vea influenciado por el scan rate del OPC. Si no se especificase el pulso, y se dejase el contacto por defecto, el incremento no sería fiel a la frecuencia especificada.

Una vez establecido el disparador de la operación, se define el bloque aritmético que se ejecutará. En este caso, se especifica el bloque ADD (suma), y se define el contenido del propio contador como primera entrada (IN1), y un 1 como segunda entrada (IN2). Esto quiere decir que el bloque cogerá el valor almacenado en el contador y le sumará 1 cada vez que el disparador se active.

Por último, se define el propio contador como contenedor del resultado de la suma. Esta operación cíclica corresponde con el comportamiento deseado de contador.

Como apunte en lo referente a los pulsos, es interesante distinguir entre pulsos positivos y pulsos negativos. Los primeros funcionan en flanco ascendente (0 a 1), mientras que los otros lo hacen en el descendente (1 a 0). Así, por ejemplo, se puede escoger el método más adecuado para cada ocasión. Un pulso positivo se puede entender como pulsar un botón, mientras que un pulso negativo se entiende como pulsarlo, para después soltarlo.

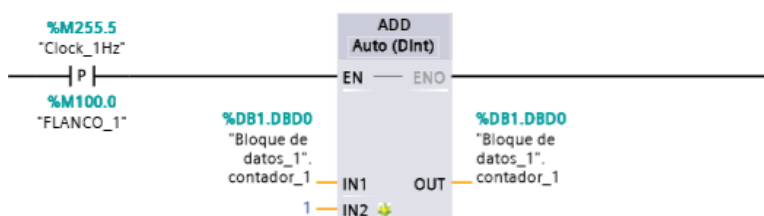


Ilustración 111: Programación de Contador

Adicionalmente, se programa también un borrado de los contadores, para el caso en que los valores de éstos fuesen demasiado altos para la simulación. Para ello, se establecen contactos sin pulso, y se utiliza el bloque lógico MOVE. Éste recibe la señal de activación (EN) de los pulsos iniciales, y escribe un 0 (IN),

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

en el contador (OUT1). Así, tenemos una manera sencilla de reiniciar los valores y hacerlos más manejables de cara a la simulación.

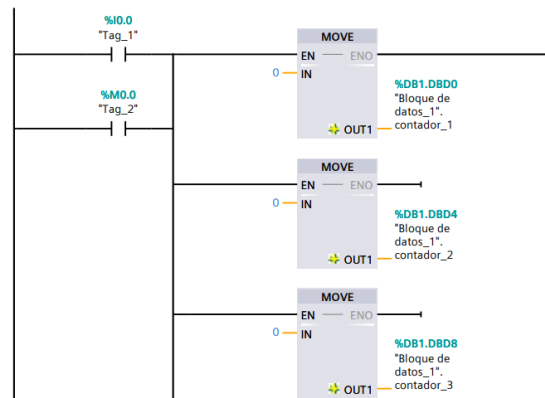


Ilustración 112: Programación de Borrado

Con la programación del PLC clara, se crea en KEPServerEX el dispositivo que lo representa, así como los tags para cada uno de los contadores (cabe decir que para que el OPC lea correctamente los valores, hay que especificar la dirección IP del PLC). Para ser más precisos, se crea un canal de Modbus/TCP Ethernet (PLC), al que se le añade un dispositivo de tipo Siemens SIMATIC S7-1200 (Device1). Además del modelo, se le asigna una ID, es decir, una dirección IP única, de manera que KEPServerEX pueda acceder por red al PLC. La dirección, en este caso, fue privada, ya que la laptop de desarrollo y el PLC de pruebas se encontraban en la misma red local. Posteriormente a la creación del canal y el dispositivo, se crean los tags del OPC (Contador_1/6). Éstos se encargarán de mapear los valores generados en el PLC mediante programación. Las direcciones utilizadas son consecutivas (DBD0-DBD4, etc.), ya que, para ahorrar espacio, se alojó cada palabra doble (DWord) una detrás de la anterior.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

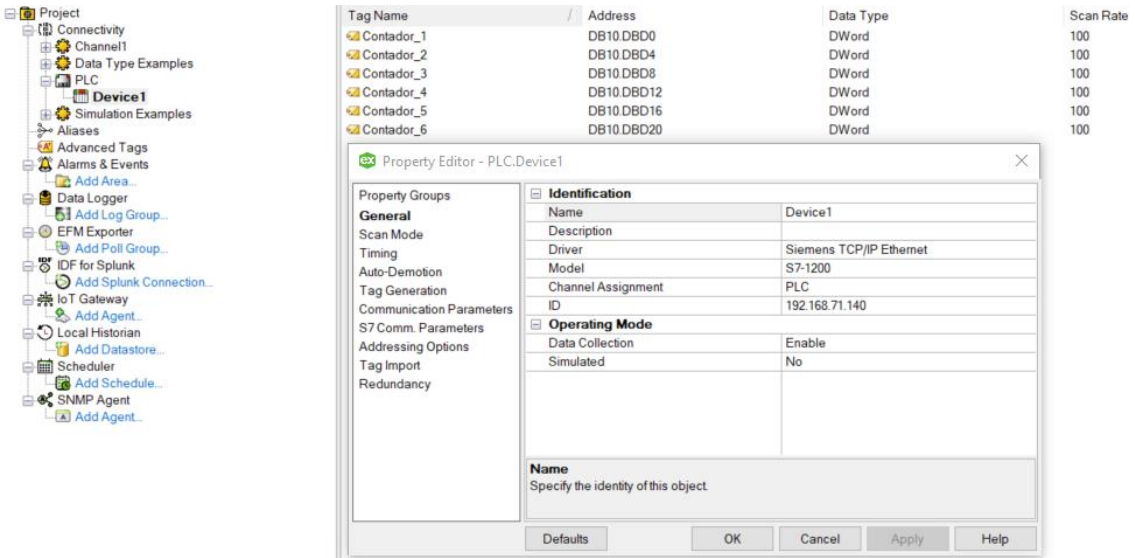


Ilustración 113: Configuración PLC (I)

Y se ejecuta un Quick Client para comprobar que los incrementos se producen según lo esperado. Se dejó trabajar al autómatas algunas horas para que los contadores tuviesen valores significativos, más útiles de cara a la simulación.

Item ID	Data Type	Value	Timestamp	Quality	Update Count
PLC.Device1_CurrentPDUSize	Word	240	08:21:59.876	Good	2
PLC.Device1_Rack	Byte	0	08:21:58.876	Good	1
PLC.Device1_Slot	Byte	1	08:21:58.876	Good	1
PLC.Device1.Contador_1	DWord	69747	08:24:22.148	Good	143
PLC.Device1.Contador_2	DWord	43592	08:24:22.148	Good	91
PLC.Device1.Contador_3	DWord	34873	08:24:21.144	Good	72
PLC.Device1.Contador_4	DWord	87183	08:24:22.148	Good	143
PLC.Device1.Contador_5	DWord	139493	08:24:22.148	Good	143
PLC.Device1.Contador_6	DWord	174367	08:24:22.148	Good	143

Ilustración 114: Configuración PLC (II)

Este testeo sirve para verificar que la arquitectura desde el nivel más bajo del despliegue, hasta la red Ethernet que alimenta la aplicación, funciona y es lo suficientemente confiable como para dedicar esfuerzos a otras áreas del proyecto. Las comunicaciones OPC son un estándar en la industria, y escogerlas dejaba poco margen a error.

KEPServerEX: Simulador Modbus y Advanced Tags

Adicionalmente a las pruebas con PLC, se realizaron pruebas con las funcionalidades software disponibles, de manera que el desarrollo pudiese abstraerse del hardware subyacente.

El software KEServerEX, en su instalación, incluye un canal predefinido con toda una serie de señales (tags) también predefinidas, que tienen como objetivo simular señales reales mediante toda una serie de funciones capaces de generar valores siguiendo un determinado patrón. La siguiente captura de pantalla lo ilustra:

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Tag Name	Address	Data Type	Scan Rate	Scaling	Description
Ramp1	RAMP (120, 35, 100, 4)	Long	100	None	Value increments by 4 from 35 to 100 every 120 ms
Ramp2	RAMP (300, 150.750000, 200.500000, -0.250000)	Float	100	None	Value decrements by 0.25 from 200.50 to 150.75 every 300 ms
Ramp3	RAMP (250, 0, 1000, 1)	Word	100	None	Value increments by 1 from 0 to 1000 every 250 ms
Ramp4	RAMP (2000, -1000, 1000, -5)	Long	100	None	Value decrements by 5 from 1000 to -1000 every 2000 ms
Ramp5	RAMP (250, -1000000, 1000000, -500)	LLong	100	None	Value decrements by 500 from 1000000 to -1000000 every 250 ms
Ramp6	RAMP (1000, 0, 1000000000, 1250)	QWord	100	None	Value increments by 1250 from 0 to 1 billion every 1000 ms
Ramp7	RAMP (1000, -1000000000, 1000000000, -5555)	LLong	100	None	Value decrements by 1 billion to -1 billion every 1000 ms
Ramp8	RAMP (1000, 150.750000, 200.500000, -0.250000)	Double	100	None	Value decrements by 0.25 from 200.50 to 150.75 every 300 ms
Random1	RANDOM (30, -20, 75)	Short	100	None	Random values from -20 to 75 that change every 30 ms
Random2	RANDOM (100, 0, 1000)	Word	100	None	Random values from 0 to 1000 that change every 100 ms
Random3	RANDOM (1000, -1000, 0)	Long	100	None	Random values from -1000 to 0 that change every 100 ms
Random4	RANDOM (1000, -999, 999)	Long	100	None	Random values from -999 to 999 that change every 1000 ms
Random5	RANDOM (1000, -1000000000, 1000000000)	Long	100	None	Random values from -1000000000 to 1000000000 that change every 100 ms
Random6	RANDOM (1000, -1000000, 1000000)	LLong	100	None	Random values from -1000000 to 1000000 that change every 1000 ms
Random7	RANDOM (1000, 0, 1000000000)	QWord	100	None	Random values from 0 to 1000000000 that change every 1000 ms
Random8	RANDOM (100, 0, 1000000)	QWord	100	None	Random values from 0 to 1000000 that change every 100 ms
Sine1	SINE (10, -40.000000, 40.000000, 0.050000, 0)	Float	100	None	Sine values between -40 and 40 at 0.05 Hz with 0 phase shift
Sine2	SINE (10, -40.000000, 40.000000, 0.050000, 180)	Float	100	None	Sine values between -40 and 40 at 0.05 Hz with 180 phase shift
Sine3	SINE (10, -40.000000, 40.000000, 0.100000, 0)	Float	100	None	Sine values between -40 and 40 at 0.1 Hz with 0 phase shift
Sine4	SINE (10, -40.000000, 40.000000, 0.100000, 360)	Float	100	None	Sine values between -40 and 40 at 0.1 Hz with 360 phase shift
User1	USER (1000, Hello.world!, This.is.a.test)	String	100	None	Sequential string values that change every 1000 ms
User2	USER (250, 15.16.23.42.4.8)	Float	100	None	Sequential float values that change every 250 ms
User3	USER (200, 1,0,0,1,0,0,1,0,0,0,1,0,1,0,0,0,0,1,0,1,0,0,0)	Boolean	100	None	Sequential Boolean values that change every 200 ms
User4	USER (1500, To display a comma, place a backslash in front of it)	String	100	None	A comma is a delimiter unless it is preceded with a backslash

Ilustración 115: Funciones de Simulación

Tal y como se muestra en la captura, se dispone de distintos tipos de funciones:

- RAMP: en régimen cíclico, incrementa un contador desde cierta valor base hasta un valor límite, sumando un cierto valor cada determinado tiempo.
- RANDOM: devuelve, cada determinado tiempo, un valor dentro del rango estipulado, con la sensibilidad marcada por el tipo de datos que maneje la función (Long, Word, etc.)
- SINE: retorna valores dentro del rango especificado de una función seno con una determinada frecuencia y desfase.
- USER: utiliza un tiempo de refresco para retornar de manera secuencial el conjunto de valores especificado como argumento

Estas funciones se asignan a un tag de la misma manera que se les asigna una dirección. Como indica la documentación de KPServer referente al driver para crear simuladores, el beneficio de estas funciones es el de encapsular en un mismo elemento lógico (el tag o la señal) la funcionalidad de direccionar de manera efectiva un espacio de memoria de la máquina con la que se desea comunicar, así como la de simular señales, que permitan parametrizar el sistema en un entorno simulado antes de desplegarlo en el ambiente de producción.

La idea para realizar este simulador es, entonces, crear un elemento Modbus Ethernet con el driver de simuladores de KPServer, alimentarlo con las funciones de simulación que ofrece dicho software, y utilizar este elemento Modbus para alimentar EnMPRO, y así poder parametrizar el sistema según las necesidades reales del despliegue. Esquemáticamente, el concepto es el siguiente:

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

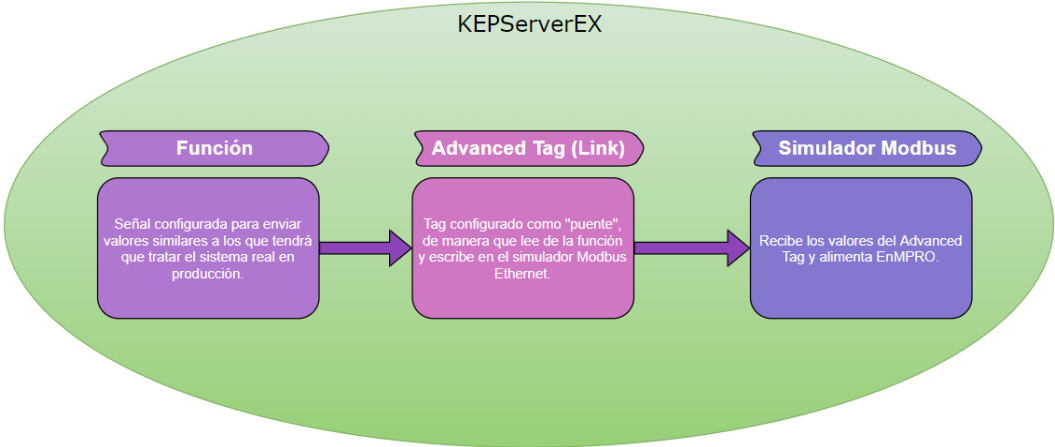


Ilustración 116: Justificación Advanced Tags

Cabe decir que este gráfico se limita a dar los detalles de cómo funciona internamente la parte de la simulación referente a KEPServerEX, sin entrar en la interacción de éste con el resto de softwares.

Para comprobar que esta arquitectura es funcional, y satisface las necesidades del desarrollo en este punto, es necesario utilizar un OPC Quick Client desplegado desde el propio KEPServerEX, y comprobar en tiempo real cómo se actualizan los valores de los tags.

El primer paso es comprobar que, en tiempo de ejecución, los tags creados mantienen sus características, es decir, que están correctamente configurados. Si se sigue el esquema anterior, tenemos que efectivamente la señal simulada retorna valores según la función especificada:

Item ID	Data Type	Value	Timestamp	Quality	Update Count
Simulation Examples.Functions.Ramp1	Long	99	10:17:22.040	Good	402

Ilustración 117: Evolución de la Señal (I)

Item ID	Data Type	Value	Timestamp	Quality	Update Count
Simulation Examples.Functions.Ramp1	Long	39	10:17:58.270	Good	437

Ilustración 118: Evolución de la Señal (II)

Si se analizan los valores de Timestamp y Update Count, se puede apreciar cómo la señal se actualiza efectivamente en el tiempo, y cómo vuelve a empezar el ciclo de suma una vez llega al final del rango establecido (basándonos en que, en un momento posterior de tiempo, el campo Value marca un valor menor que el que marca en la lectura más temprana). Se puede afirmar entonces que esta señal es útil para comprobar la conveniencia de la arquitectura propuesta, por lo que el siguiente paso es chequear que el

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Advanced Tag en modo Link interconecta satisfactoriamente la función con el driver Modbus Ethernet.

Para realizar la comprobación, se tiene que navegar dentro del OPC Quick Client hasta la sección donde se agrupan todas las Advanced Tags que están funcionando en el sistema. Allí, su campo Value mostrará el mapeo que ese Tag lleva a cabo, y se podrá comprobar si es el deseado:

Item ID	Data Type	Value	Timestamp	Quality	Update Count
AdvancedTags.SimulatorTag0	String	Simulation Examples.Functions.Ramp1 == MB Simulator.MB.Test0	10/10/36.838	Good	1

Ilustración 119: Advanced Tag

En el campo Value se puede apreciar como este Advanced Tag iguala, es decir, mapea, la salida de SimulationExamples.Functions.Ramp1 con la entrada de MBSimulator.MB.Test0. Esto quiere decir que el “puente” entre los dos Tags, la interconexión entre las dos señales se ha realizado de manera satisfactoria. En este caso, es interesante entender que el Update Count se mantiene a 1. El motivo es que mientras que el valor mapeado que pasa “a través” de este enlace cambia en cada ciclo de escaneo (Scan Rate), el Advanced Tag en sí no varía. Para completar esta parte de la comprobación, nos referiremos a KEPServerEX, y visualizaremos la configuración del Advanced Tag.

Ilustración 120: Configuración de Advanced Tag (Link)

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

La captura de la pantalla de configuración muestra el funcionamiento interno de esta señal. Como ya se ha explicado, es del tipo Link Tag, y mapea `SimulationExamples.Functions.Ramp0` con `MBSimulator.MB.Test0`. Además, podemos ver cómo se le podría haber asignado un Dead Value si fuese necesario, y haber cambiado las políticas de actualización. En este caso, se confió en la configuración por defecto, por lo que el Advanced Tag siempre está funcionando (Trigger con valor Always), realiza el enlazado cada segundo (Link Rate con valor 1000 milisegundos), y enlaza a cada cambio en la señal de entrada (Link Mode con valor On Data Change of Input Tag).

El último paso es comprobar que la parte final del mapeo se realiza correctamente. Para ello se comprueba en el OPC Quick Client que el Tag creado dentro del simulador de Modbus se actualiza correctamente con los valores proporcionados mediante el Advanced Tag.

Item ID	Data Type	Value	Timestamp	Quality	Update Count
MB Simulator.MB.Test0	Word	95	11:18:43.234	Good	42

Ilustración 121: Comprobación del Flujo de Datos (I)

Item ID	Data Type	Value	Timestamp	Quality	Update Count
MB Simulator.MB.Test0	Word	63	11:18:57.252	Good	56

Ilustración 122: Comprobación del Flujo de Datos (II)

En las capturas de pantalla anteriores se puede ver cómo el Tag se actualiza con el paso del tiempo (vuelven a ser importantes los campos Timestamp y Update Count), y podemos, decir que el flujo de datos se realiza correctamente de un extremo a otro.

Para una configuración más precisa, se podría haber comprobado cuáles eran los ciclos de escaneo y actualización del resto de señales, pero se consideró que no era fundamental para comprobar el correcto funcionamiento de la arquitectura de simulación, y se dejó como opción de mejora durante el proceso de parametrizado.

Item ID	Data Type	Value	Timestamp	Quality	Update Count
Simulation Examples.Functions.Ramp1	Long	63	11:41:03.149	Good	84

Ilustración 123: Comprobación Final (I)

Item ID	Data Type	Value	Timestamp	Quality	Update Count
MB Simulator.MB.Test0	Word	63	11:41:03.320	Good	407

Ilustración 124: Comprobación Final (II)

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Hubiese sido plausible que, dependiendo de los tiempos de escaneo y enlazado, se observase una diferencia evidente entre ambos extremos de la comunicación, debido al decalaje temporal producido por los parámetros de la simulación, pero en este caso, y como las imágenes ilustran, pese a notar una diferencia de Timestamp de algunas décimas de segundo, se podría decir que la actualización de los valores se produce prácticamente en tiempo real.

EnMPRO Web: Dashboard de widgets

Los dashboard en web resultan una manera intuitiva y sencilla de visualización, con diferentes objetos y opciones.

La versión 7.1 de SIMATIC Energy Manager PRO incluye una interfaz web, que permite la gestión del sistema de ingeniería desde cualquier punto que esté conectado a la red donde habita el servidor. Esta interfaz tiene las mismas funcionalidades que el cliente de escritorio, pero las ofrece de manera más intuitiva y agradable a la vista.

Para realizar las pruebas se crea un canal en KepServerEX, y se crean dos tags con las funciones de simulación que provee el mismo software.

Tag Name	Address	Data Type	Scan Rate	Scaling	Description
Sim1	RAMP (1000, 0.000000, 10000.000000, 1.000000)	Float	100	None	Rampa
Sim2	SINE (10, -40.000000, 40.000000, 2.000000, 0)	Float	100	None	Seno

Ilustración 125: Tags creados

Sim1 es un simple contador que se reinicia al llegar a un límite superior, mientras que Sim2 dibuja una función sinusoidal. Con un QuickClient revisamos que los tags están dando valores válidos:

Item ID	Data Type	Value	Timestamp	Quality	Update Count
rampas.SimCounter.Sim1	Float	708	12:26:19.879	Good	11
rampas.SimCounter.Sim2	Float	39.1942	12:26:18.877	Good	3

Ilustración 126: Tags en el Quick Client

Una vez comprobamos que las señales son funcionales, pasamos a abrir EnMPROWeb en un navegador, y a crear un nuevo dashboard que nos permita visualizar los valores de los tags creados:

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES



Ilustración 127: Widget Dashboard

La gráfica de la izquierda representa la función sinusoidal Sim2, mientras que el medidor radial de la derecha muestra el valor del contador Sim1. Los cuadrados dentro de la propia gráfica permiten mostrar histogramas y datos históricos, pero en esta simulación lo pasaremos por alto.

Cabe recordar que según la infraestructura de EnMPRO explicada con anterioridad, el software tiene que alimentarse de los datos de otros softwares llamados Acquisition, encargados de funcionar como punto de salida de datos de cada planta de producción hacia el EnMPRO corporativo.

<input type="checkbox"/>	Name ^	Description ^	Status ^	Host ^	IP address ^	GUID ^
<input type="checkbox"/>	h_Local_PC	local	Active	EnMPRO	192.168.1.104	bc8dc8aa-1f2d-45ca-b9db-b81ebac138f4

Ilustración 128: Acquisition para la Simulación

Se pueden crear instancias de Acquisition a través del cliente web, y gracias al campo IP que éstas tienen asociado, vincularlos con el equipo encargado de enviar los datos de cada planta. Posteriormente se asocia cada tag (denominados aquí como Data Point), con su correspondiente Acquisition, para completar el mapeo y asegurar que los datos mostrados se corresponden con las lecturas de la planta escogida.

El software ofrece diversos tipos de widgets, o elementos gráficos para mostrar datos e información. Es importante conocer cada tipo, de manera que se pueda interpretar de manera correcta su contenido y aprovechar al máximo el valor añadido que éstos pueden aportar.

Gauge (medidor) y semáforo

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

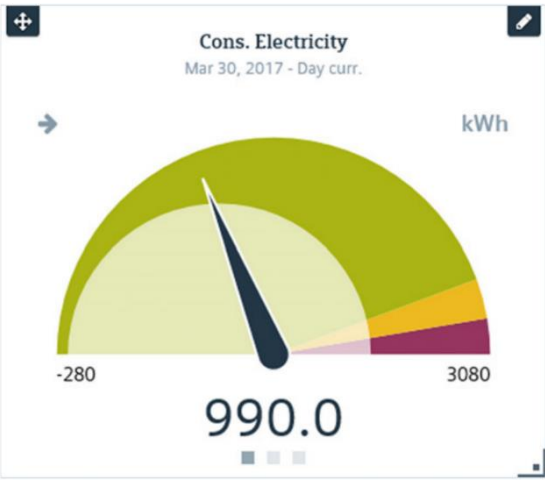


Ilustración 129: Gauge (medidor)



Ilustración 130: Semáforo

Estos dos elementos son dos de los más comunes a la hora de mostrar si una lectura se encuentra dentro de cierto rango (y el valor de la lectura en el caso del medidor). Son, además, dos maneras simples y altamente familiares de mostrar datos a operarios que no necesariamente han de estar familiarizados con formas más complejas de gráfico.

Min	<input type="text" value="0"/>	Min-Warning	<input type="text" value="0"/>
Max	<input type="text" value="2800"/>	Max-Warning	<input type="text" value="2600"/>
Min-Range	<input type="text" value="0"/>	Max-Range	<input type="text" value="3000"/>
Violation:	<input type="checkbox"/>	Warning:	<input type="checkbox"/>
	<input type="checkbox"/>		<input type="checkbox"/>
Value:	<input type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>		<input type="checkbox"/>

Ilustración 131: Configuración medidor/semáforo

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Pese a sus similitudes, ambos tienen diferencias que hay que saber apreciar. El semáforo simplemente muestra dentro de qué rangos está situado el valor medido, y la tendencia que está siguiendo. El medidor permite, además de visualizar el valor concreto de la lectura, apreciar de manera visual cómo de cerca o lejos se está de los límites definidos en la configuración, además de la tendencia actual y la unidad usada en la representación.

Chart

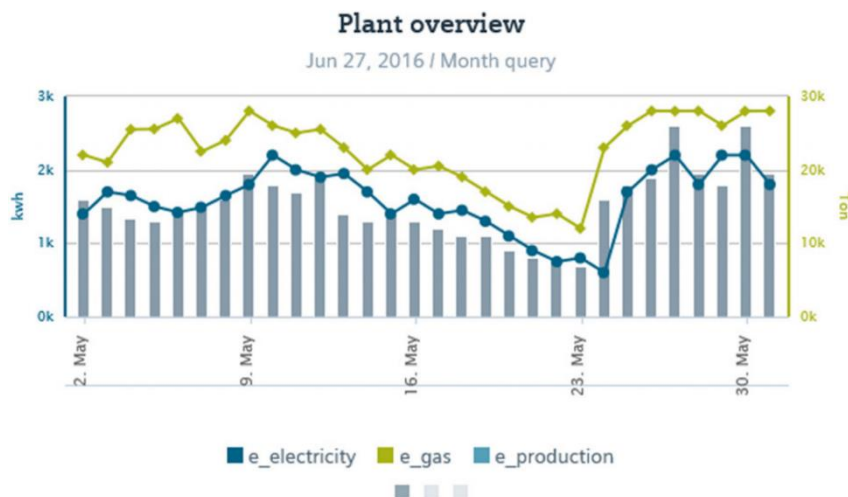


Ilustración 132: Chart

Otro método muy familiar es el de una gráfica como la de la imagen sobre estas líneas. Este tipo de representaciones multivalue son especialmente convenientes para comparar valores tomados en distintos momentos en el tiempo, ya que permite mostrar hasta 10 puntos de datos distintos, distribuidos en hasta 5 ejes Y y 2 ejes X. Por ejemplo, se podría mostrar simultáneamente los mismos datos con una semana de desfase, y detectar anomalías o tendencias de interés para el proceso de análisis.

Pueden aparecer en forma de línea, barras o área, y se pueden definir límites de la misma manera que con los medidores, que mostrarán la aceptabilidad de un dato al seleccionarlo en el widget.

Pie Chart

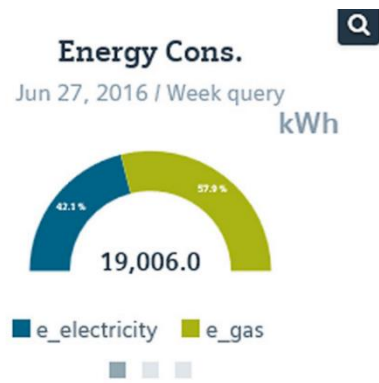


Ilustración 133: Pie Chart

Aunque no recomendado según algunas directrices de diseño gráfico, los pie chart son útiles para mostrar distribuciones sobre un concepto general, como la participación de los consumos de electricidad y gas en el consumo general.

Table

The figure is a table titled "Energy Consumption" with a subtitle "Feb 29, 2016 / Week query". The table has three columns: "Timestamp", "d_counter kWh", and "d_temp °C". It contains eight rows of data. Below the table are three small gray squares.

Timestamp	d_counter kWh	d_temp °C
01:00:00	4,872.00	39.94
01:00:00	2,628,881.00	19.72
01:00:00	7,109,444.00	19.32
01:00:00	11,524,063.00	18.77
01:00:00	15,854,385.00	17.35
01:00:00	18,395,124.00	6.62
01:00:00	19,611,604.00	4.95

Ilustración 134: Table

Una tabla es una buena opción a la hora de mostrar series temporales de varios puntos de datos a la vez. Dispone de diversas páginas si no se pueden mostrar todos los valores a la vez, y de diversas pestañas en las que hay información y estadísticas más detalladas.

Valores



Ilustración 135: Valores

Este widget permite mostrar los valores de una manera sencilla y directa, pudiendo cambiar el nombre de las variables en el propio gráfico o desde Energy Manager PRO. Ofrece información más detallada en otras pestañas.

Reporte

Para no tener que cambiar de ventana para visualizar reportes, este widget permite mostrar reportes existentes y cambiar el periodo de tiempo mediante controles. También se puede abrir como fichero Excel o PDF, e iniciar los cálculos en caso de que el reporte no exista previamente.

Texto/Imagen

Newsletter 

The last month energy consumption could be reduced by **5.6 %**.

This is equivalent to a reduction of **125 tCO₂!**

Ilustración 136: Texto

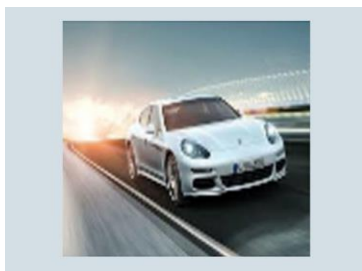


Ilustración 137: Imagen

Estos dos elementos permiten, por un lado, mostrar texto enriquecido con hipervínculos a otros sitios web, y por el otro, imágenes, que con la debida configuración, pueden actuar como botones de navegación hacia otros dashboards.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Mapa de calor (Heatmap)

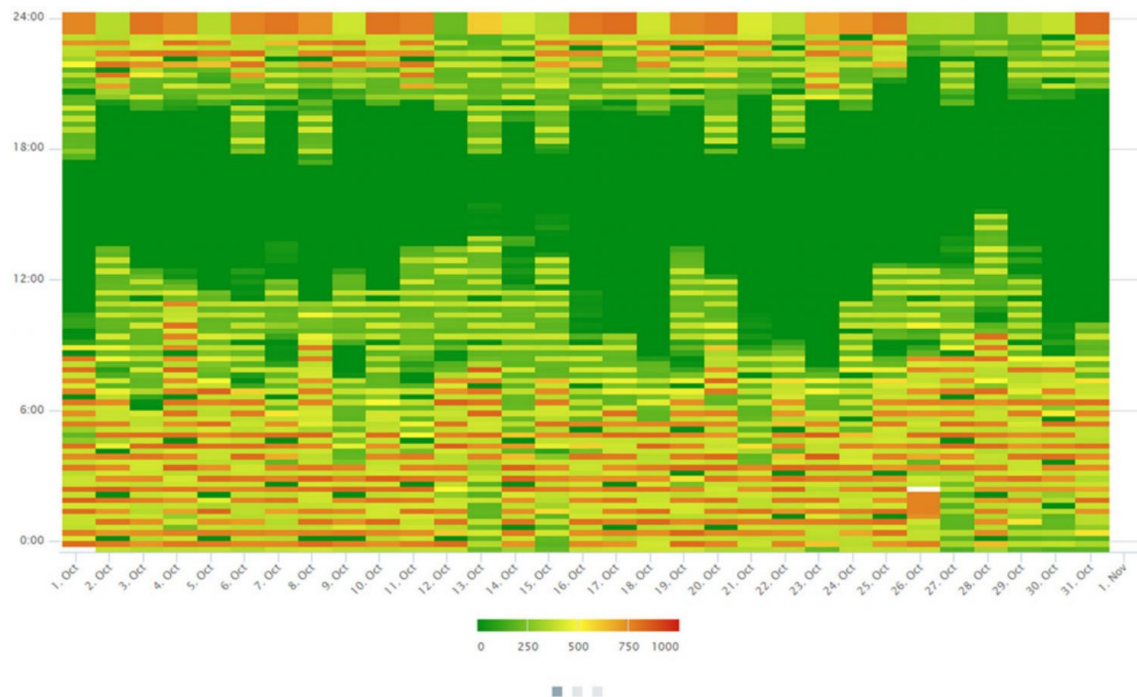


Ilustración 138: Heatmap

Dada una variable, un mapa de calor permite mostrar un histórico con dos niveles de sensibilidad. Por ejemplo, en el de la figura, el eje horizontal muestra días, mientras que el vertical muestra horas del día. De esta manera, con la leyenda debajo del mapa, podemos saber qué valor tenía la variable en cada momento de cada día dentro del rango configurado.

Diagrama Sankey



Ilustración 139: Diagrama Sankey

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Con estos diagramas se pueden asociar varias variables, de manera que se puede visualizar el flujo de energía entre las diferentes partes del proceso. En la imagen se puede ver como el consumo de gas se relaciona con el funcionamiento del compresor, y éste con el de la hervidora. Si se selecciona un enlace se muestra información sobre el consumo, y en vistas secundarias se pueden visualizar las relaciones entre enlaces y sus valores numéricos.

Link Name	From Node	To Node	Value	Primary Energy
Strom_Ein	Strom_Ein	Compressor	-0,18 kWh	-0,18 kWh
Gas_Ein	Gas_Ein	Compressor	0,05 kWh	5,61 kWh
Druckluft_out	Compressor	Boiler	0,27 kWh	5,37 kWh

Ilustración 140: Relaciones Sankey

Por último, permite mostrar información a nivel de enlace, como las entradas y salidas, la diferencia entre ambas, y la desviación relativa.

Node Name	Input	Output	Difference	Relative
Strom_Ein	0,00 kWh	-0,18 kWh	0,18 kWh	100,00 %
Gas_Ein	0,00 kWh	5,61 kWh	5,61 kWh	100,00 %
Compressor	5,43 kWh	5,37 kWh	0,06 kWh	1,03 %
Boiler	5,37 kWh	0,00 kWh	5,37 kWh	100,00 %

Ilustración 141: Nodos Sankey

Dinámica de Reportes

Aquí se explica cómo se pasa de datos almacenados a reportes personalizados, que podrían ser incrustados en otras herramientas.

No hay que olvidar que una de las finalidades de este sistema será la de presentar los datos de manera que representen de manera intuitiva y conveniente la información que se puede extraer de ellos. Más allá de las funcionalidades más técnicas del sistema, éste tiene que tener también una faceta más informativa, y es por eso que tiene que existir toda una arquitectura que permita generar reportes y facturas. Con esto tenemos que el cumplimiento de los objetivos de este sistema es integral, y se satisface a todas las partes de un hipotético entorno corporativo: los técnicos manipulan las mediciones y la gerencia recibe datos de calado económico-empresarial.

La herramienta escogida para realizar a cabo las tareas de Reporting será SSRS (SQL Server Reporting Services). Mediante Visual Studio, se puede crear un proyecto como si de una aplicación de escritorio se tratara, y poblarlo con reportes que se pueden diseñar en el propio IDE, o importar de proyectos existentes.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Para el caso que nos ocupa, tenemos que los proyectos de Reporting Services son desarrollados suponiendo que serán desplegados en un ambiente de producción, con un origen de datos que alimentará los reportes, y que en la red existirá la configuración adecuada para que el personal de la industria pueda acceder a los reportes cuando lo necesiten. A continuación, se muestra un ejemplo de reportado de información:

Primero hay que definir un origen de datos desde el cual los reportes puedan alimentarse. En este caso se ha creado una base de datos de ejemplo que contiene datos similares a los que se podrían encontrar en un entorno de producción real:

	IDMaquina	UnidadesProducidas	VelocidadNominal
1	0	500	10
2	1	600	25
3	2	350	15
4	3	600	30

Ilustración 142: Base de Datos de Producción

En esencia, esta tabla nos muestra las unidades producidas por cada máquina, así como la velocidad nominal de esta. En tanto que una simulación, no se ha tenido en cuenta las unidades de los datos en la base de datos, dado que el objetivo, más que ofrecer datos reales, es dar una idea de cómo funciona la dinámica de reportado.

La organización de esta clase de proyectos de reporting suele ser similar en la mayoría de casos. El proyecto cuenta con uno o varios Orígenes de Datos, a partir de los cuales se generan diversos Conjuntos de Datos, y finalmente los propios reportes. La necesidad de tener distintos Conjuntos de Datos radica en diferenciar las necesidades de cada reporte: partiendo de que todos los datos provienen de una Base de Datos única, es posible que un reporte solo necesite ciertos campos de información, otro necesite campos distintos, y un tercero requiera de la base de datos completa. Para el caso que nos ocupa, solo se ha generado un Conjunto de Datos:

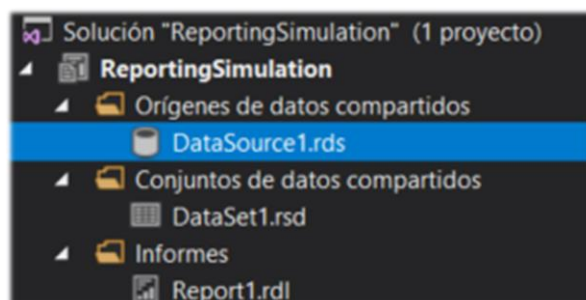


Ilustración 143: Conjunto de Datos

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Como se puede observar, creamos DataSet1 a partir de DataSource1, y de esta manera podemos alimentar Report1.rdl.

A continuación de crean todos los reportes necesarios. Esta tarea siempre tiene que tener en cuenta el propósito del proyecto de Reporting, es decir, la naturaleza del negocio de dónde provienen los datos, y la manera más conveniente de mostrar la información por pantalla. De esta manera se pueden diferenciar reportes de información de planta, de evolución de consumos, o de producción. Son solo ejemplos de la gran diversidad de información que los reportes pueden aportar al personal de las organizaciones.

La imagen siguiente muestra un reporte de ejemplo generado con los datos de la tabla creada:

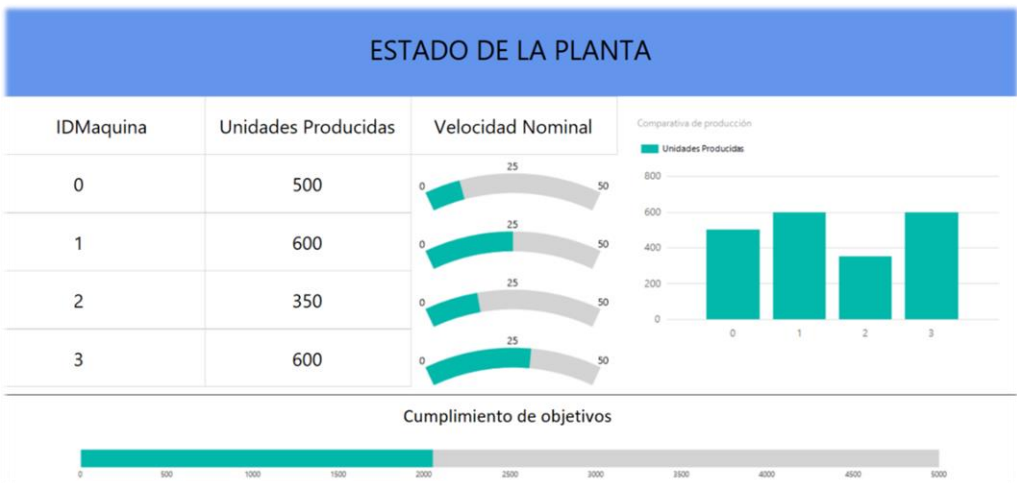


Ilustración 144: Reporte

En el reporte se puede apreciar que las máquinas quedan clasificadas según su identificación. La producción a nivel de máquina es representada con un número, mientras que la velocidad nominal se representa con un medidor radial. La ventaja de SQL Server Reporting Services es que permite mostrar la información de diferentes maneras, tal y como se ve en la imagen.

Gracias al medidor radial, nos podemos hacer una idea de cómo de cerca está cada máquina de trabajar a velocidad máxima, gracias a los valores mínimos y máximos del propio medidor. De igual manera, podemos comparar de manera visual la producción de cada máquina gracias al gráfico situado a la derecha, y el medidor lineal de la parte inferior nos informa de cuánto queda para cumplir el objetivo, que en este caso es el de fabricar 5000 unidades del producto.

SSRS también permite interactuar con procedimientos almacenados en la base de datos, así como establecer parámetros que, o bien serán introducidos en tiempo de ejecución al querer ver el reporte, o serán manejados internamente según las especificaciones del diseñador del proyecto. La mayoría de objetos

tienen propiedades que permiten configurar disparadores para todo tipo de acciones, como navegar a otro reporte, o a una URL especificada.

Aunque existe la posibilidad de desarrollar mediante

Metodología de trabajo

Aclaración sobre cómo se desarrollará este proyecto en cuanto a organización, y algunas consideraciones adicionales.

En este proyecto, le estudiante que desempeña el rol de desarrollador pertenece a un equipo dentro del departamento, pero colabora con gente del mismo departamento y e otros. Dentro del propio departamento y fuera de él, compañeros y compañeras especializadas ayudarán al estudiante. En el caso que nos ocupa el jefe de departamento hace las veces de cliente del proyecto, y transmitirá los requerimientos de manera periódica.

En cuanto a metodologías de trabajo, cualquiera del tipo Agile cuadra bastante con lo que es el desarrollo de este proyecto, debido a la adaptabilidad que ofrece frente a cambios en tiempo de desarrollo. En éste caso, la memoria escrita conforma una documentación más elaborada de lo que se podría esperar de otros proyectos que usasen esta metodología. Se pondrá más énfasis en la colaboración cliente-desarrollador que en el cumplimiento riguroso de contratos y fechas límite.

Se combinará de manera flexible el proceso de aprendizaje con la realización de tareas que se consideren viables en ese momento. El jefe de departamento dará feedback continuo, que ayudará a solucionar los problemas que vayan surgiendo, y a dar forma al futuro del proyecto.

Herramientas de seguimiento

¿Cómo se monitorizará el desarrollo?

Con el objetivo de tener un seguimiento del proyecto, la principal herramienta serán las presentaciones periódicas que se realizarán, de manera que os responsables puedan ver los avances en la implementación. Mediante BitBucket, los responsables tendrán constancia de las solicitudes de envío, y después de una revisión, aceptarlas. Otras funcionalidades como la creación de ramas facilitarán la identificación de desviaciones en el decurso previsto del proyecto.

Herramientas de validación

¿Cómo se comprobará que el desarrollo va por buen camino?

Los principales métodos para validar el desarrollo del proyecto serán las reuniones periódicas realizadas con los responsables, y las revisiones realizadas a través de BitBucket. Se creará también un entorno simulado, donde probar el correcto funcionamiento, partiendo de una serie de premisas predefinidas y similares a las de un despliegue real.

Posibles obstáculos

¿Hay alguna fuente de retrasos prevista?

- Curva de aprendizaje:
 - La mayoría de tecnologías que se usarán resultan nuevas, por lo que hay que tener en cuenta el tiempo y los recursos necesarios a invertir en el proceso de entrenamiento.
- Bugs:
 - Toda aplicación en desarrollo tiene errores que se han de solucionar.
- Calendario:
 - Por razones imprevistas como salidas a eventos de diversa índole, problemas técnicos o similares, es posible que la planificación inicial se modifique para adecuarla a las nuevas necesidades.

Planificación general

Explicación y justificación del desarrollo del proyecto tanto en tiempo como en recursos.

Este proyecto pertenece a la categoría de TFGs en modalidad B, es decir, con Convenio de Colaboración Educativa con una empresa, que aloja al estudiante en prácticas durante la realización de su proyecto. La duración de estas prácticas es de 735h.

Inicialmente se esperaba que éste proyecto fuese defendido en enero de 2020, pero gracias a cambios en el convenio, se podrán realizar las 735h de manera que pueda ser defendido en octubre de 2019. Es por este cambio que se mostrarán ambas planificaciones (original y final).

La duración media de éste tipo de proyectos es de 6 a 12 meses, pero teniendo en cuenta las necesidades impuestas por el calendario, se acotarán tanto el alcance como los requisitos, para adecuarlo a los tiempos disponibles.

Descripción de tareas

Segmentación del trabajo en unidades comprensibles y justificación de cada una.

Aprendizaje, entrada en contexto y comprensión completa del negocio y la arquitectura

Aun tratándose de una empresa informática, y de aplicar conceptos que me son familiares, a raíz de los requisitos propuestos, he tenido que utilizar herramientas que son nuevas para mí. Nunca antes había utilizado Wonderware, Kepware, Reporting Services o Visual Studio. En la mayoría de casos he tenido que estudiar la documentación de la compañía según lo necesitaba, y en otros, la empresa ha designado un tutor para que me instruyese. También he tenido que entender la arquitectura general de los servicios que ofrece la empresa en la que hice las prácticas, ya que he tenido que basar mi propuesta en pautas similares. Además, les tuve que pedir documentación, tanto propia como de terceros, que me ayudase a entender cómo encajaría mi proyecto en el mercado, es decir, en el ámbito de la informática industrial, y asistir a reuniones con el jefe de departamento para que

me informase acerca de la mecánica de trabajo y demás aspectos del negocio.

El objetivo final del proceso de desarrollo de este proyecto es similar al que suele tener la empresa en los suyos: desarrollar una herramienta que pueda monitorizar procesos (en este caso consumos energéticos) e historizar y tratar datos según las necesidades concretas del despliegue.

Estas soluciones requieren la instalación de un motor de base de datos como SQL Server, que permita almacenar los valores de las lecturas y organizarlas. Posteriormente hay que saber encajar el software de gestión energética escogido sobre la arquitectura física de captación instalada en planta. Antes de decantarme por desarrollar una aplicación propia que sirviese como complemento, barajé toda una serie de alternativas, como usar exclusivamente EnMPRO, y cada una tenía unos requerimientos distintos, que podían hacer cambiar de manera importante la planificación del proyecto. Además, la documentación de estos softwares no siempre es demasiado clara, por lo que el proceso de instalación y prueba de cada uno de ellos podía alargarse más de lo que me convenía.

La metodología Agile me permitió flexibilizar todas estas sub-tareas, de manera que cumplía con los plazos que se me marcaban, y compaginaba el trabajo de aprendizaje con la implementación de otras.

Obtención de requisitos y análisis

Una vez se halla un punto de encuentro entre los intereses del estudiante y las necesidades de la empresa, se inicia un proceso de negociación. Al final de este proceso, el responsable en la empresa extiende al estudiante los requisitos del proyecto que va a elaborar, de manera que quede encauzado dentro de su actividad de negocio. Con la propuesta sobre la mesa, se tiene que barajar los detalles técnicos necesarios para la implementación, y posteriormente documentarlos para tener una base teórica en la que fundamentar el desarrollo.

En una primera instancia, se barajó el uso de equipo más costoso, que permitía la construcción de una arquitectura que se extendía desde la planta a la gerencia de la empresa, y hacía posible la historización de datos a largo plazo sin necesidad de DataLoggers adicionales. Luego se revisó la propuesta y la idea de utilizar equipos de otro fabricante, más económicos, tomó fuerza. Éstos, a diferencia de los anteriores, limitaban su alcance a la planta de producción, y requerían de equipos adicionales si se deseaba tener históricos de datos a largo plazo.

Cabe decir que, en esta etapa, está incluido todo el proceso de negociación de requisitos y objetivos del proyecto, que conforman el trabajo previo al proceso de implementación, una vez se tiene el permiso de los responsables.

En resumen, los principales recursos necesarios para esta etapa son:

- Tiempo de negociación y búsqueda de opciones.
- Adquisición de requisitos de parte del responsable en la empresa.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

- Negociación posterior de ajuste y redacción de propuesta original.

Instalaciones físicas

En conjunción con los compañeros especialistas en automatismos y electricidad, se procede a levantar la infraestructura física necesaria para hacer posible la captación de señales. Se construye la red que utilizarán los contadores, con sus correspondientes routers y switches, y se establecerán pasarelas y conversores entre las distintas conexiones. El estudiante no formará parte activa del proceso de construcción, pero sí que depende de su progreso, y estará al tanto de la arquitectura utilizada, además de colaborar en la investigación de tecnologías y arquitecturas para el despliegue.

Los recursos principales utilizados en esta etapa son los siguientes:

- Tiempo de construcción de la infraestructura.
- Posibles desplazamientos.
- Inversión de capital para la adquisición de equipos (convertidores, conectores, etc.)
- Técnicos especialistas en automatismos e instalaciones eléctricas.

Diseño del sistema de gestión de recursos

De manera paralela a la instalación de la infraestructura física en las plantas, en las oficinas de la empresa se desarrollará el sistema de gestión, es decir, se desarrollará la herramienta de gestión y monitorización de recursos energéticos, adecuándola a los requisitos del proyecto.

Una vez se concluye el proceso de simulación en entorno seguro, y se haya concluido el proceso de instalación de contadores, se podrá desplegar la solución en un ambiente de producción.

Los recursos principales de esta tarea son los siguientes:

- Implementación del sistema de gestión.
- Inversión de capital en la adquisición de software.
- Equipos informáticos.
- Informáticos que lleven a cabo la tarea.

Entrenamiento del personal y entrega de la solución

Después de implantar con éxito la solución, comprobando que se cumplen satisfactoriamente todos los requisitos, hace falta entrenar al personal en su uso y mantenimiento. De esta manera este proyecto no acaba en su entrega, si no que los efectos de su desarrollo pueden permanecer en la empresa para posteriores proyectos, y sirve al estudiante para probar el rol de tutor en vez del de tutelado.

Estos son los recursos utilizados:

- Tiempo invertido en instrucción y formalización de entrega

Pruebas y soporte

Al final del proceso, se dará soporte a la empresa, y en el caso que surgiese cualquier problema o duda acerca del funcionamiento del sistema, el

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

estudiante se encargará de solucionarla. Además, durante esta tarea se estudiarán las diversas maneras de demostrar el trabajo realizado en la defensa del TFG.

La lista de recursos asignados a esta tarea son los siguientes:

- Tiempo de búsqueda de opciones de cara a la defensa del proyecto.
- Tiempo de servicio técnico para solucionar problemas y dudas.

Tiempo estimado

Descripción a nivel de tiempo del desarrollo del proyecto, y justificación de cambios frente a la planificación original.

Durante todo el proyecto se ha optado por una filosofía de trabajo próxima a las directrices de las Agile, por su flexibilidad y adaptabilidad a cambios. A continuación, se muestra la planificación original, para después describir el decurso real del proyecto y justificar las diferencias.

Planificación original

Tarea	Duración
Aprendizaje	Continuo
Requisitos y Análisis	1 mes → 80h
Instalaciones, sistema y .NET	2 meses → 160h
Entrenar y Entregar	1 mes → 80h
Pruebas y Soporte	2 meses → 160h

Ilustración 145: Planificación Temporal Original

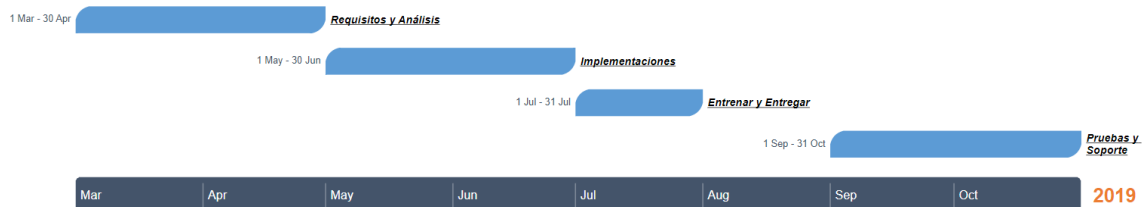


Ilustración 146: Gantt Original

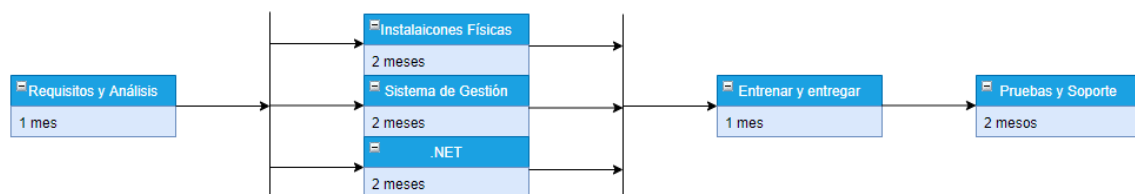


Ilustración 147: Dependencias Originales

Desarrollo real

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Tarea	Duración
Aprendizaje	Continuo
Requisitos y Análisis	3 meses → 281h
Instalaciones, sistema	3 meses → 324h
Entrenar y Entregar	1,5 meses → 171h

Ilustración 148: Planificación Temporal Definitiva

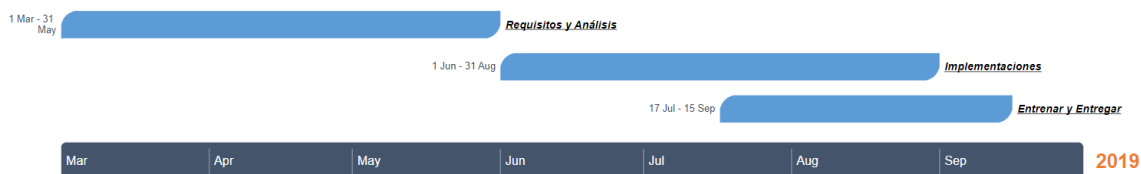


Ilustración 149: Gantt Definitivo

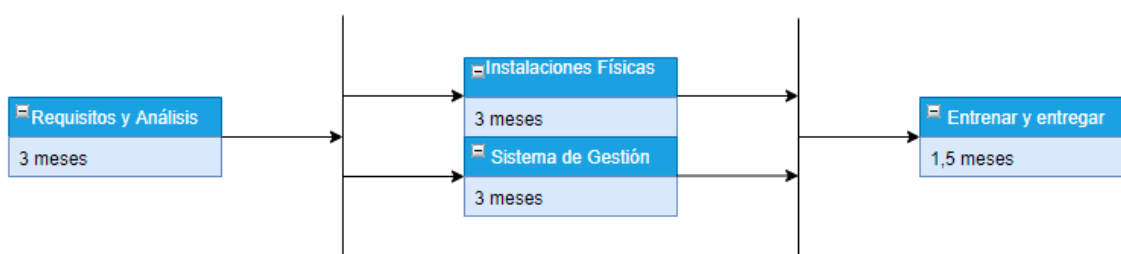


Ilustración 150: Dependencias Definitivas

Justificación y explicación de variaciones

Como la planificación definitiva refleja, se ha optado por tres grandes bloques de trabajo:

- Requisitos y análisis
- Implementaciones
 - Instalaciones Físicas
 - Sistema de Gestión
- Entrenar y Entregar

En la sección correspondiente se explica en qué consiste cada una de las tareas. Este apartado se dedicará a justificar las variaciones en el plan de trabajo.

Primeramente, es necesario señalar que, debido a una modificación en el convenio de prácticas, la organización temporal del proyecto sufrió cambios notables. De la idea de defender este TFG en enero de 2020, se pasó a defenderlo en el turno de octubre de 2019. Esto hizo necesario un ajuste en las horas de prácticas en la empresa, realizándose con varios meses menos de margen. Para hacer este adelanto factible, tuve que replantearme el alcance práctico del proyecto, y se eliminó la parte de Pruebas y Soporte, que no era plausible dentro de los nuevos términos. Parte del tiempo que se dedicaba a esta tarea se desplazó a la de Entrenar y Entregar.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Por otro lado, se desestimó la opción de desarrollar aplicaciones en .NET encima del sistema de gestión, ya que éste ya tiene funcionalidades de visualización y análisis de todo tipo. Además, la propuesta de aplicación en .NET que había sobre la mesa confiaba en el sistema que la empresa utilizaba, por lo que además de aprender a desarrollar la aplicación en sí, hubiese hecho falta familiarizarse con dicho sistema, para ver cómo llevar a cabo la integración. La gran cantidad de tiempo que llevaría hizo que se considerase inviable. En vez de esto, se propuso crear una aplicación .NET independiente, que interactuase con el servidor de datos OPC, y ofreciese funcionalidades de gestión y monitorización energética, y fuese capaz de exportar datos a EnMPRO, para que éste pudiese aplicar sus funcionalidades estadísticas y contables más avanzadas.

Por último, pese a tener una idea más o menos clara del objetivo del proyecto, los responsables de la empresa tuvieron inconvenientes, y la entrega de requisitos se demoró más de lo previsto. El tiempo se aprovechó para asentar las bases tanto teóricas como prácticas que me permitirían llevar a cabo este proyecto, lo que corresponde a la tarea Aprendizaje, llevada a cabo durante la totalidad de las prácticas.

Alternativas y planes de acción asociados

Explicación de los desvíos sufridos de planificación sufridos, y medidas que se tomaron para corregirlos.

Problemas de fechas

El cambio en los términos del periodo de prácticas, y la consiguiente variación de la fecha de defensa de la memoria comportaron una reestructuración del contenido del proyecto. De la misma forma, la demora en la obtención de requisitos provocó un desplazamiento en el tiempo de todas las demás tareas del proyecto, obligando a tomar decisiones como descartar el periodo de Pruebas y Soporte.

Dificultades con las herramientas

La poca claridad de algunas herramientas a la hora de instalarlas o utilizar funcionalidades avanzadas obligaba a invertir tiempo en solventar errores. Algunos de los medios utilizados estuvieron cerca de quedar descartados a causa de estas cuestiones.

Falta de experiencia

La falta de experiencia en proyectos de gran envergadura, orientados a entornos de producción, y el desconocimiento de las herramientas, hizo que se tuviesen que corregir las expectativas del proyecto. En su fase más temprana, la idea fue llevar a la web el sistema de automatización que ofrecía la empresa, pero la compleja estructura de éste, en conjunción con la inviabilidad de los requisitos y la falta de manejo por parte del estudiante, hicieron que el proyecto virase en otra dirección.

Presupuesto del proyecto

Explicación de los gastos asociados al proyecto, y de los cambios sufridos respecto del presupuesto original.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Presupuesto original

Hardware

Producto	Precio	Amortización
Toshiba ThinkPad T570	1250€	Vida Útil 5 años → 104,17€
Contadores	975€ - 1125€	Empresa: AI
Instalación	12500€	AI Entregar

Ilustración 151: Presupuesto Hardware Original

Software

Software	Precio	Amortización
EnMPRO	7650€	Empresa: AI Entregar

Ilustración 152: Presupuesto Software Original

Recursos Humanos

Rol	€/h	Horas	Total
Desarrollador	8	240h	1920€
Project Manager	8	160h	1280€
Tester	8	160h	1280€

Ilustración 153: Presupuesto RRHH Original

Presupuesto final

Hardware

Producto	Precio	Amortización
Toshiba ThinkPad T570	1250€	Vida Útil 5 años → 104,17€
SIEMENS 7KM PAC3200	559,42€ * 60uds = 33565,2€	10.588,39€/año → 3,17 años
MOXA MGATE MB3180	219€ * 2uds = 438€	138,17€/año → 3,17 años
Instalación	12500€	AI Entregar

Ilustración 154: Presupuesto Hardware Final

Software

Software	Precio	Amortización
EnMPRO (licencias, configuración, etc.)	11300€	Empresa: Al Entregar

Ilustración 155: Presupuesto Software Final

Recursos Humanos

Rol	€/h	Horas	Total
Desarrollador	25	324h	8100€
Project Manager	45	281h	12645€
Tester y Entrenador	25	171h	4275€

Ilustración 156: Presupuestos RRHH Final

Justificaciones y diferencias

Empezando con el Hardware, finalmente el precio de los contadores fue notablemente mayor de lo previsto. Originalmente los requisitos propuestos por la empresa fueron orientativos, y al concretarse el modelo de contador, y las unidades, el presupuesto subió significativamente. En estos proyectos, la instalación queda amortizada al entregar el proyecto, ya que se suele incluir en el primer pago al desplegar. Por último, el laptop mantiene la misma amortización, ya que, en el cómputo global funciona el mismo número de horas que en la planificación original.

El Software se encareció, debido a que las licencias de este tipo de aplicaciones van en función del número de señales que se tratan.

En cuanto a Recursos Humanos, en el presupuesto inicial se igualó el salario por hora al acordado por el convenio universitario. Más adelante, se decidió que, aunque la remuneración del estudiante sí que es de 8 €/h, en un proyecto real debería ser más alta. Por lo tanto, y siguiendo la filosofía de este proyecto, en parte consistente en hacer un estudio y planificación de un despliegue real, se acordó que en el presupuesto de RRHH debían figurar números más próximos a la realidad. De aquí la diferencia visible en el presupuesto de RRHH definitivo.

Descripción de las herramientas

Esta sección se encarga de dar una idea intuitiva de las principales herramientas software utilizadas en este proyecto.

SIMATIC Energy Manager PRO



Ilustración 157: Logo SIEMENS

Software desarrollado por SIEMENS que ofrece todo un framework de gestión y monitorización del consumo energético a nivel corporativo. En un nivel más bajo, se alimenta de softwares replicados llamados Acquisition a nivel de planta, encargados de enviar al EnMPRO corporativo los datos recogidos para su posterior tratamiento y análisis.

Su objetivo es ofrecer una solución escalable para el cumplimiento de la normativa ISO 50001-1 sobre sistemas de gestión energética. Éste es un sistema tradicional de gestión basado en Planificación-Acción-Revisión-Acción, destinado a reducir el consumo energético de cualquier tipo de organización, independientemente de su tamaño, ámbito o localización geográfica. La certificación de esta normativa por auditores acreditados asegura la correcta implantación del conjunto de directrices, y refuerza la imagen de la organización de cara a la competencia y a los inversores.

Manual oficial:

https://support.industry.siemens.com/cs/attachments/109742442/simatic_energy_manager_pro_operating_manual-en-US-en-US.pdf

WonderWare



Ilustración 158: Logo WonderWare

Software propiedad de AVEVA, originalmente desarrollado por Schneider Electric que permite construir la red lógica de automatización de una industria. Entre sus principales funcionalidades se encuentran, además, la capacidad de servir como suite de automatización completa para la organización, aunque es común el uso de sus partes, ya sea su sistema de historización de datos (Historian), su entorno de desarrollo de la arquitectura (Archestra IDE) o sus capacidades de SCADA o de interfaz para HMI, entre otros.

Actualmente, AVEVA divide las funcionalidades de WonderWare en las siguientes categorías: HMI/SCADA, IIM, MES/MOM. A continuación, se muestran algunas de estas funcionalidades bajo su categoría madre:

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

- HMI/SCADA:
 - Integración de Dispositivos: interfaz de comunicación entre dispositivos.
 - InTouch: software de visualización de procesos en HMI.
 - System Platform: plataforma SCADA para una estandarización a nivel corporativo.
- Industrial Information Management:
 - Historan: base de datos de alto rendimiento.
 - Insight: solución para gestión de información destinada a usuarios finales y constructores de equipos.
 - Intelligence: simplificación del mapeo de las lecturas con los KPIs de la compración de los OEE de diferentes localizaciones.
- MES/MOM:
 - Manufacturing Execution System: sistema de monitorización integral de las materias primas al producto final.
 - Batch Management: solución de procesos de batch para automatización.
 - Insight Performance: solución basada en cloud para la eficiencia del equipo de manufactura y el uso de los datos recogidos.

Web del producto:

<https://www.wonderware.com/>

KEPServer



Ilustración 159: Logo Kepware-KEPServer

Software propietario de Kepware encargado de implementar la comunicación OPC en los equipos de planta. Mediante la creación de túneles, es capaz de transferir los datos entre dos o más equipos, de manera segura, y sin necesidad de conexiones VPN o softwares adicionales como Microsoft COM o DCOM.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES



Ilustración 160: Concepto KEPServerEX

Además de sus capacidades básicas de OPC, cuenta con toda una serie de capacidades adicionales, que dan un valor añadido al uso de este software. A continuación, se listan algunas:

- Tags avanzados: señales configurables que permiten no solo la adquisición de datos de las máquinas, sino que hace posible establecer condiciones de disparo para la adquisición de datos, la totalización de eventos o el cálculo de medidas de un grupo de sensores.

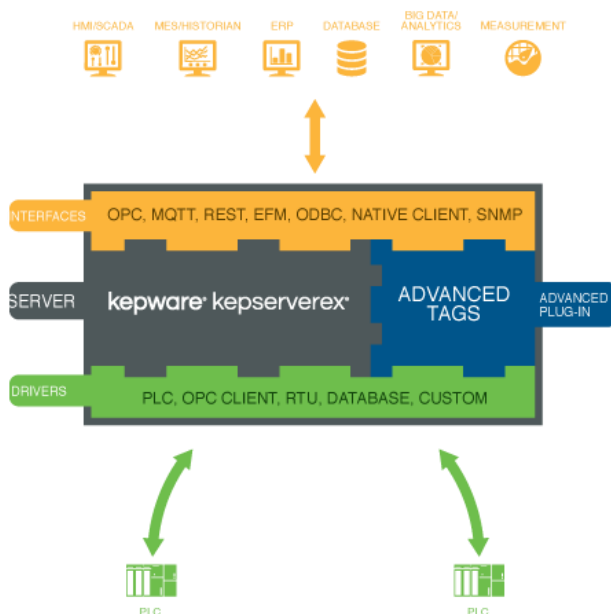


Ilustración 161: Tags Avanzados

- Alarmas y Eventos: mediante la configuración del módulo, permite utilizar la información proveniente de los dispositivos para servir los datos de las

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

alarmas, y visualizar éstas y otros eventos a través de un cliente del sistema.

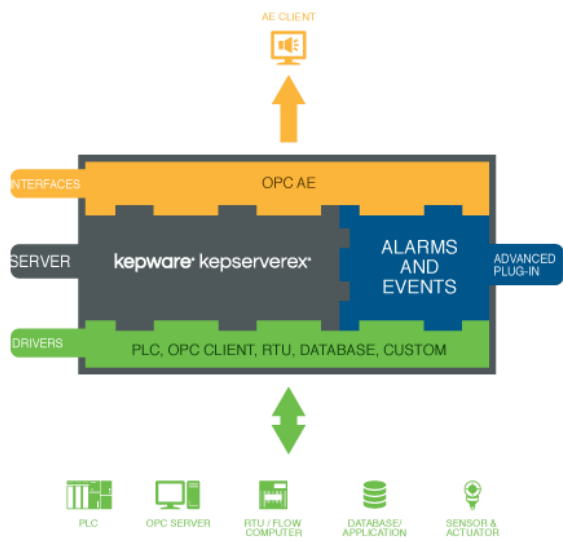


Ilustración 162: Alarmas y Eventos

Web oficial:

<https://www.kepware.com/en-us/products/kepservex/>

Arquitectura de la solución

Aquí se ilustra el diseño de la solución tratada, y se explica qué función cumple cada elemento dentro de la estructura general.

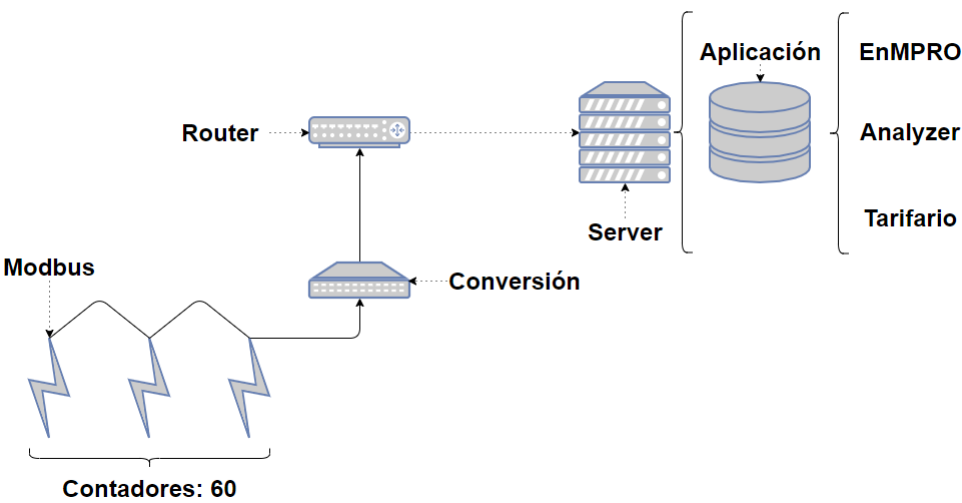


Ilustración 163: Arquitectura de la Solución

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

A continuación, se definirá cada parte, y se enlazará con la siguiente, de manera que al final se obtenga la lógica de la arquitectura general.

Mercado: Grandes fabricantes de sensores/contadores

Breve estudio de mercado en búsqueda de grandes fabricantes.

Electricidad: Socomec



Ilustración 164: Socomec

Empresa de originaria de Francia, especializada en equipos de medición, conversión y conmutación energética, así como control de consumo. Dispone de soluciones integrales para la generación, control y renovación de la energía eléctrica, así como productos dedicados al ámbito médico, al ferroviario, y las energías renovables.

Desarrollaron el concepto DIRIS Digiware, según el cual el sistema se hace modular, y permite al cliente la máxima flexibilidad posible en el diseño de una solución personalizada. Se utilizan pasarelas de conversión para centralizar todo el flujo de mediciones en un único aparato, y enviar los datos a la red Ethernet vía Modbus TCP.

Resulta interesante, para el propósito de éste proyecto, la gama de sensores de intensidad de que dispone, en la cual se pueden encontrar dispositivos tanto de núcleo abierto como cerrado, y compatibles con su sistema DIRIS Digiware.

Fluidos: Endress & Hauser



Ilustración 165: Endress & Hauser

Grupo de empresas dedicado al desarrollo de equipos para la automatización de procesos, así como sistemas de medición, y soluciones software. En su web, ofrecen sugerencias de equipos que podrían ser del interés de una industria específica, basándose en los datos introducidos a través de una interfaz.

Dado que su principal atractivo son los sensores de medición de caudales, presión, temperatura y nivel, resulta una opción plausible para cualquier compañía que maneje fluidos, y necesite tener bajo control los parámetros de éstos.

Contadores y red Modbus

Explicación del modelo de contador utilizado, y de la red Modbus que los interconecta.



Ilustración 166: SENTRON 7KM PAC3200

El contador utilizado pertenece a la gama SENTRON de SIEMENS. Es la alternativa con mejor relación prestaciones-coste, y resulta adecuado para las necesidades de este proyecto. Es capaz de medir tanto la energía aparente, como la activa y la reactiva, admite módulos de expansión, y soporta tensiones de entrada de hasta 690V. También admite entradas analógicas mediante la instalación de las expansiones correspondientes, y es el primero de la gama que admite hacer interfaz con Energy Manager PRO.



Ilustración 167: Modbus

Creado por Modicon (actualmente Schneider Electric), Modbus es un protocolo de comunicaciones pensado para el entorno industrial, basado en la filosofía de maestro-esclavo. Según esto, un dispositivo maestro realiza polling secuencial a sus esclavos, de manera que estos les envíen los datos que han leído. Para ello, el protocolo cuenta con un modelo de datos específico, adaptado a su propósito, y que tiene en cuenta la diferenciación de accesos, dependiendo del rol del dispositivo.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Bloque de Memoria	Tipo de Datos	Acceso de Maestro	Acceso de Esclavo
Bobinas	Booleano	Lectura/Escritura	Lectura/Escritura
Entradas Discretas	Booleano	Solo Lectura	Lectura/Escritura
Registros de Retención	Palabra Sin Signo	Lectura/Escritura	Lectura/Escritura
Registros de Entrada	Palabra Sin Signo	Solo Lectura	Lectura/Escritura

Ilustración 168: Bloques de Modelo de Datos

Todo el modelo de datos está representado en un espacio de direcciones de 65536 elementos (2^{16}), y depende totalmente de cómo decida organizarlo el esclavo. Se puede imaginar como una cuadrícula, en la que las filas vienen determinadas por los números de registro, y las columnas por el tipo de este registro. De esta manera, un maestro puede pedir el valor de cierta combinación fila-columna, o de manera análoga, sobrescribir el dato que en ella se encuentra.

#	Coils	Discrete Inputs	Input Registers	Holding Registers
1	1		200	50
2	0			68
3	0			1000
4	0			
5	1			
6	1			
7	0			
8	0			
9	0			
10				
11				

Ilustración 169: Representación Direcciones Modbus

Pese a ser visualmente intuitiva, a la hora de configurar, la utilización de la metáfora de matriz resulta poco conveniente, ya que hace que la especificación de registro y tipo de registro se alargue innecesariamente. Por ello, se utiliza una nomenclatura que utiliza prefijos para identificar el tipo de registro, y números para especificar la dirección. De esta manera, se tiene un dígito para identificación de tipo, y cinco dígitos para direccionar cualquiera de las 65536 direcciones. Por ejemplo, el registro de retención número 1 se direccionaría con el 400001, mientras que la bobina 4001 se direccionaría con el 004001.

Bloque de Datos	Prefijo
Bobinas	0
Entradas Discretas	1
Registros de Entrada	3

Ilustración 170: Prefijos Modbus

Los equipos Modbus pueden comunicarse de diferentes maneras (Ring, Daisy Chain, etc.), pero por norma general, una red Modbus multipunto cuenta con un cable principal (trunk), y posiblemente varios cables derivados (se establecen terminaciones de línea por cuestiones de impedancia). Para el caso que nos ocupa, la topología aplicada es la llamada Daisy Chain, mediante la cual todos los esclavos se conectan a la línea trunk que sale del maestro. Esta arquitectura tiene la ventaja de simplificar el cableado, pero depende del número de esclavos que pueda soportar el PLC maestro, así como del tráfico que puede soportar la línea de comunicación. El tercer aspecto a tener en cuenta es la latencia de los paquetes, determinada por el número de veces que éstos tengan que pasar por el proceso de store&forward en cada equipo. La fórmula general para calcular esta latencia es la siguiente:

$$\text{Lat. Total} = (\text{Size Paquete} * 8) / \text{Tasa} + \text{Lat. Forwarding}$$

Aunque no hay directrices estrictas sobre las limitaciones de estas redes, en entornos industriales, un principio básico es que la red es tan rápida como su dispositivo tan lento. Concretamente, se espera que el RTT (Round Trip Time), en la red sea de menos de 5ms. Para ello, se suele limitar el número de equipos directamente conectados al trunk a 32. Si se decide localizar más equipos, se tendrían que tener en cuenta las implicaciones, en el momento de diseñar la red. En nuestro caso, como el número de contadores es de 60, se decidió crear dos redes Modbus de 30 unidades cada una, y unir las a la red mediante los adaptadores de los que se hablará más adelante.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Climatix POL63x Modbus-Master

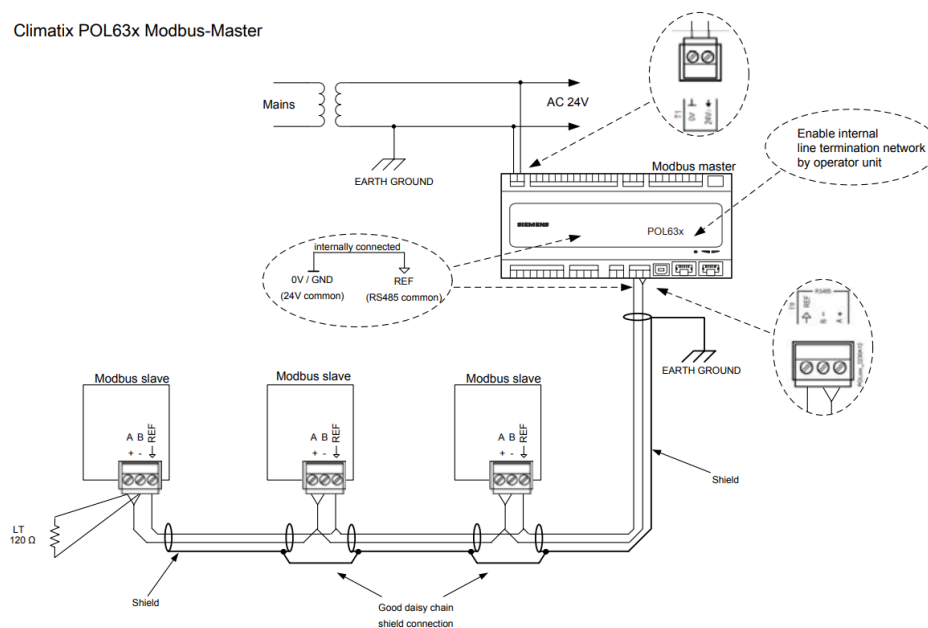


Ilustración 171: Topología Daisy Chain

Conversión Serie a Ethernet

Modbus es el método de comunicar los contadores, pero ¿cómo pasamos al Ethernet que necesitaremos más adelante?



Ilustración 172: Conversor Serial a Ethernet

Protocolos como Modbus resultan muy convenientes a nivel bajo, es decir, en el ámbito de los contadores y los PLC, pero cuando se pasa a niveles más altos de la arquitectura, carece de las funcionalidades de otros como Ethernet. Es por eso que después de crear la infraestructura de contadores, se lleva las señales hacia un conversor, que las agregará y las convertirá a protocolo Ethernet, haciendo posible su tratamiento como otra parte más del tráfico de red de la planta. Gracias a esta conversión, pasamos de tener dos subredes

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

independientes, a convertirlas en tramas que TCP/IP soporta, y hacerlas pasar por un router, de manera que obtenemos un único flujo de datos entrantes al servidor donde se aloja la aplicación.



Ilustración 173: Ejemplo de Conversión Serial-Ethernet

Existen toda una multitud de conversores, y se diferencian tanto por el número de puertos RS-232 que incluyen como por su ámbito de aplicación o funcionalidad:

- Por número de puertos:
 - De 1 puerto: 1 puerto RS-232 a 1 puerto Ethernet.
 - De 2 puertos: 2 puertos RS-232 a 1 puerto Ethernet, pudiendo usar los primeros de forma simultánea o alternativa.
 - Multipuerto: 1/2/4/8 puertos RS-232 a 1 puerto Ethernet, pudiendo usar los primeros de forma simultánea o alternativa.
 - Modelos High-Port: 16/24/48 puertos RS-232 a 1 puerto Ethernet, pudiendo usar los primeros de forma simultánea o alternativa. Adaptados para montaje en rack para una mejor organización de los equipos de red.
- Ámbito o funcionalidad:
 - Modelos Industriales: de construcción más resistente, resistencia calórica más elevada, y prestaciones software más avanzadas. Pueden estar adaptados para instalación en carril DIN, e incluir encriptación del tráfico enviado a la red Ethernet.



Ilustración 174: Conversor Industrial RS-232 a Ethernet

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

- Modelos Inalámbricos (Wireless): envían el tráfico Ethernet de forma inalámbrica, haciéndolo conveniente en entornos de difícil cableado. Suelen ser más costosos y tener menos puertos RS-232 que sus variantes por cable.



Ilustración 175: Conversor Inalámbrico RS-232 a Ethernet

- Modelos PCB (Printed Circuitry Board): sin protección externa, la disposición en circuito impreso facilita su integración en caso de querer incluirlo en equipos existentes. Pueden permitir modificaciones en su chip y software.



Ilustración 176: Conversor PCB RS-232 a Ethernet

Como se ha podido apreciar, RS-232 es ampliamente aceptado de conexión serie. El estándar dicta qué voltajes tienen que utilizarse, las frecuencias y otras características eléctricas, así como la especificación física de los pines, la función de cada circuito en los conectores, y las propiedades mecánicas de éstos. Más allá de directrices en el ámbito de lo físico, el estándar deja a las partes de la comunicación definir formatos y velocidades de transmisión, con la recomendación de no exceder los 20000 baudios (bits por segundo), ya que no está pensado para eso.

No hay que olvidar el propósito original de RS-232, que es conectar un terminal con un módem. Partiendo de esto, hay diferentes limitaciones que es necesario

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

considerar. Por ejemplo, las grandes variaciones de voltaje requeridas para cambiar de positivo a negativo, o los 25 pines del conector original.

Analyzer (Acquisition), EnMPRO

Justificación de ambos en el contexto de la solución

Una vez llegan al servidor, los datos están listos para ser almacenados y tratados. Siguiendo la arquitectura de EnMPRO mencionada en otras secciones de esta memoria, los datos pasan por un software tipo Analyzer llamado Acquisition. El propósito de éste es comprimir los datos y realizar un primer análisis de éstos, para después enviarlos a Energy Manager PRO, que los almacenará a largo plazo y realizará las tareas de análisis avanzado necesarias para el proceso de Mejora Continua.

Aunque la idea de ésta separación Acquisition-EnMPRO es contar con un número indeterminado de instancias del primero para una única instancia del gestor general, este proyecto solo tiene en cuenta una planta de producción. Esto no es un problema, ya que, gracias a las opciones de configuración por interfaz de la herramienta, el proceso de añadir analizadores de planta al sistema se simplifica.

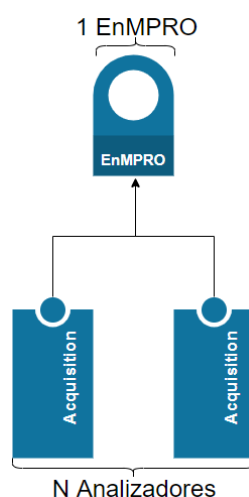


Ilustración 177: Acquisition y EnMPRO

Aplicación: Tarifario

Explicación del objetivo y de la metodología de desarrollo

Esta parte del sistema tiene el objetivo de recoger los datos, organizarlos convenientemente y exportarlos, de manera que, a la hora de generar facturas propias, se pueda utilizar información válida. Esto dará legitimidad a la contabilidad propia, y protegerá al usuario de fraudes en el cobro de facturas. Adicionalmente, permite visualizar en tiempo real la evolución de los consumos, ver cómo afecta eso al consumo energético, y generar un fichero que resuma la situación general en cuanto a consumos y gastos.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Los precios de la energía varían a lo largo del día, a razón de seis precios diarios, lo que deja cada jornada dividida en 6 bloques de 4 horas cada uno. La idea es estructurar en forma de fichero CSV estos datos, y serializarlo desde el código, de manera que tanto EnMPRO como cualquier otra herramienta, que acepte este formato, pueda utilizarlo. Cada día se publican los precios del día siguiente, así que el proceso extracción-inyección de tarifas se puede pensar como una tarea susceptible de ser automatizada.

Aunque de interfaz y propósito simples, este software cumple con el propósito de poner al alcance de operarios, a priori poco cualificados, las altas prestaciones de los OPC, y de poder aplicar éstas a un propósito intuitivo y de importancia. Creando interfaces accesibles entre la mano de obra y los complejos sistemas subyacentes, se simplifica el uso de las nuevas tecnologías, y se justifica la inversión que necesitan, en tanto que representan un gran valor añadido.

APIs

Las APIs (Application Programming Interface) son un conjunto de definiciones de funciones, protocolos de comunicación y herramientas, destinadas a abstraer a quien programe de la arquitectura subyacente, y exponerle solo las acciones y objetos que necesite. De esta manera, es más sencillo integrar sistemas y aplicaciones ya desarrollados en nuevos proyectos, ya que solo es necesario estudiar la documentación de la API para entender cómo pedirle la información, y cómo la retorna.

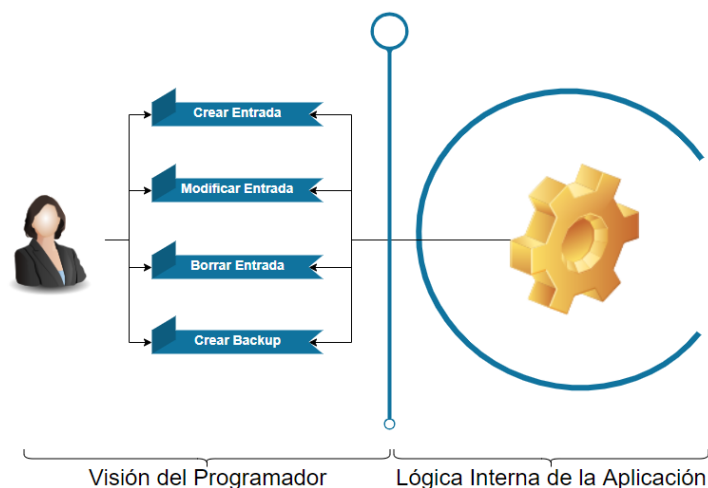


Ilustración 178: Concepto de API

Mediante una API gubernamental, se hace la consulta correspondiente, y ésta devuelve la planificación de precios de la energía. Posteriormente solo es necesario organizar la información recibida e inyectarla en el sistema.

Aunque fundamental para el correcto funcionamiento de la solución, no es de esperar que se realice la consulta de tarifas justo cuando el sistema necesite actualizar los precios. Es por eso que se optó por realizar comunicaciones de

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

manera asíncrona: no merece la pena bloquear la aplicación hasta recibir los datos.

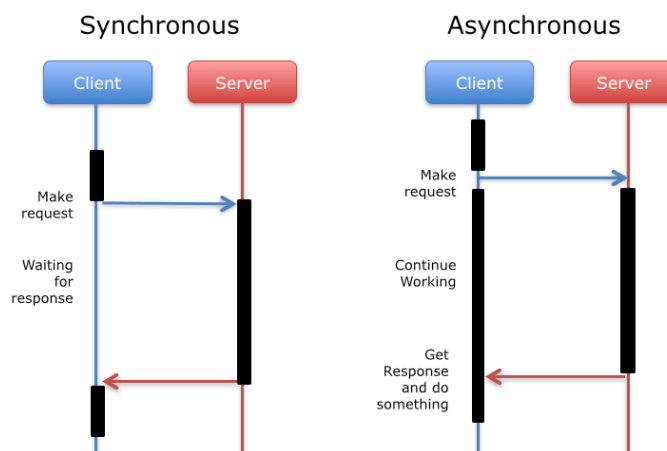


Ilustración 179: Comparación Síncrono-Asíncrono

Ficheros CSV

Los ficheros CSV son una forma sencilla de representar datos organizados en forma tabular. Aunque su nombre (Comma-Separated Values) sugiere que los valores se separan por comas, la realidad es que éstos pueden ser separados por otros caracteres, o incluso espacios o tabulaciones. Es por esto que, comúnmente, los programas que aceptan importaciones de datos vía ficheros CSV permiten definir el carácter a interpretar como delimitador.

[illegible]

Ilustración 180: Correspondencia Excel-CSV

Dependiendo de la naturaleza de los valores, es posible que contengan caracteres que coincidan con el definido como delimitador. Para evitar incoherencias a la hora de importar los datos, se pueden definir caracteres o secuencias de escape, o también editar manualmente o con un script el contenido del CSV para adaptarlo.

Implementación del software

Detalle del desarrollo del código y justificación de las decisiones tomadas.

Framework y tipo de proyecto

Breve comentario del framework utilizado, y explicación del tipo de proyecto. En las imágenes, por norma general, se mostrará la especificación de clases y métodos.

En la empresa que aloja el convenio educativo al que estoy adherido, entre otros, se utiliza mayormente el framework .NET de Microsoft, mayormente aplicado a la programación en C# con Visual Studio. Era natural, entonces, que el conocimiento que adquiriré desempeñando prácticas en dicha empresa me hiciese escoger las mismas herramientas para realizar mi proyecto.

En tanto que un entorno de programación maduro y extensamente documentado, .NET es muy común a la hora de desarrollar aplicaciones complejas, y el amplio abanico de posibilidades que ofrece lo hacen factible tanto para crear aplicaciones de escritorio, como webs, juegos, drivers o apps para dispositivos móviles. En mi caso, me interesaba que fuese una aplicación de escritorio. Podría haber desarrollado una solución para la web, pero hubiera sido una dificultad añadida para la cual hubiera necesitado más tiempo y experiencia.

Antes de entrar en el detalle del concepto del programa, es importante aclarar la lógica que utiliza Visual Studio para jerarquizar los ficheros que conforman la solución.

- Por encima de todo está la solución. Esta sirve como contenedor de todas las funcionalidades que se supone están pensadas para solventar el objetivo del software. De ahí el nombre de "Solución" (para un problema a resolver).
- En un nivel inferior se encuentra una colección de "Proyectos". Visual Studio ofrece plantillas para distintos tipos de proyectos: desde una biblioteca o un servicio, hasta un proyecto de Windows Forms o una página web.
- También puede haber recursos independientes, como iconos o ficheros auxiliares.

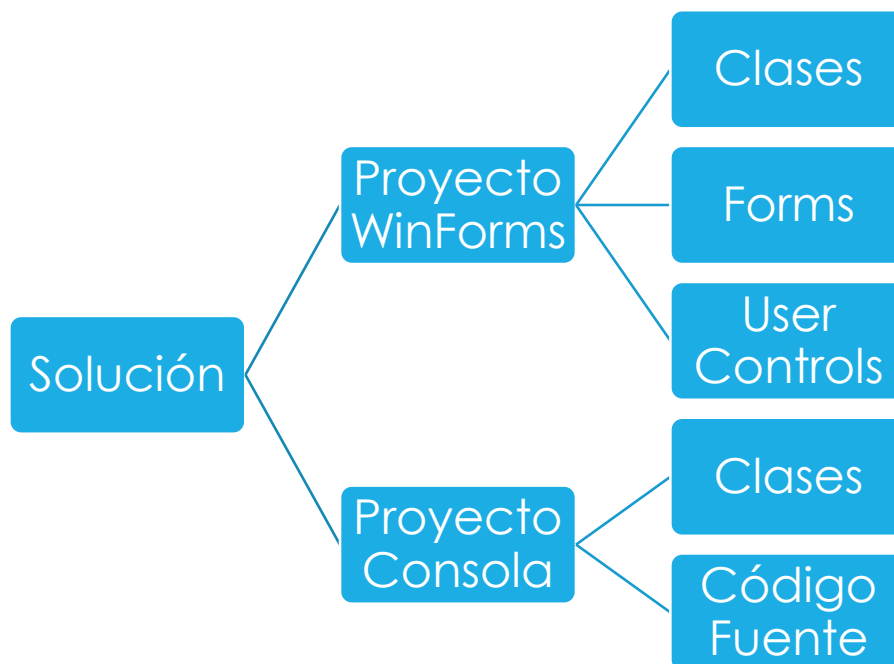


Ilustración 182: Ejemplo de Solución en VS

La aplicación desarrollada, mayormente es un proyecto de Windows Forms. En este tipo de proyectos, comúnmente se tiene un programa de inicio, utilizado para mostrar una de las ventanas creadas, así como el propio conjunto de ventanas, las clases de datos utilizadas, o los User Controls diseñados. Estos últimos son unidades funcionales discretas, destinadas a realizar tareas específicas, como rellenar los datos de una estructura, o leer la entrada de datos de un usuario. La ventaja del uso de User Control es que, si en tiempo de diseño, se tiene en cuenta la reutilización de código, se pueden diseñar controles genéricos, que se puedan incrustar en diferentes ventanas a lo largo de la aplicación. En la aplicación que nos ocupa, el desarrollo del UserControl podría haberse evitado, ya que la complejidad del proyecto hubiese justificado prescindir de él, pero representaba un concepto interesante que decidí explorar.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

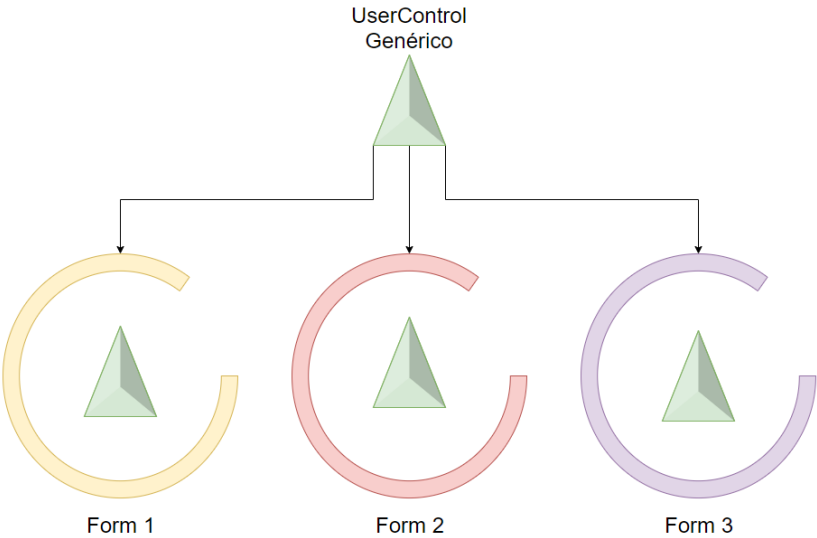


Ilustración 183: Concepto UserControl

The screenshot shows a configuration window titled 'Atributos de la Configuración'. It features a table with columns for 'Hora Inicio', 'Hora Fin', and 'Precio'. Below the table are buttons for 'Descartar Valores', 'Guardar Tramos', 'Nombre del Vector', 'Unidad de Medida', and 'Guardar Vector'. The window also includes a section for 'Añadir Vector a Fuentes' and 'Borrar Fuentes', with fields for 'Nombre', 'Directorio de Guardado', 'Fecha de Vigencia', and 'Divisa'. The window is titled 'Atributos de la Configuración'.

Ilustración 184: UserControl en la Aplicación

En cuanto a cómo fluye el uso de la aplicación, opté por un funcionamiento intuitivo y sencillo. Al iniciarse, permite escoger si se desea empezar de cero con una nueva configuración, o si, por el contrario, se desea cargar una ya existente y trabajar en base a ella. También permite entrar en una pantalla de monitorización, en caso de que no se quisiera editar ningún tipo de configuración. Más adelante en este documento, se hará un recorrido por las pantallas existentes.

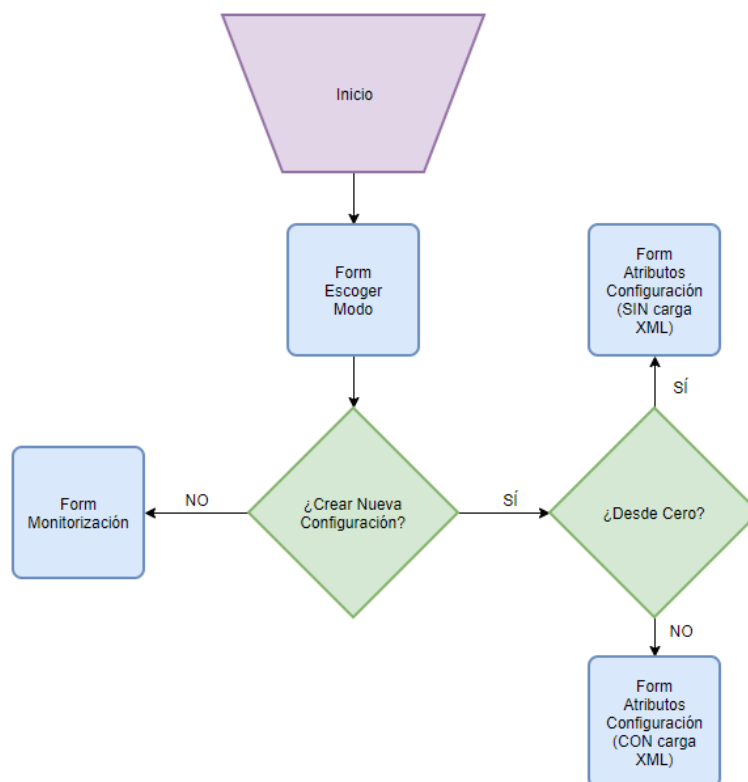


Ilustración 185: Flujo de funcionamiento

Jerarquía de clases

Justificación del modelo de datos escogido

Para facilitar el desarrollo y la modularidad del código, opté por una estrategia bottom-up, es decir, empecé por las clases más sencillas, y fui componiendo con estas las más complejas, en los niveles superiores de la jerarquía. De esta manera, cree la clase Tramo, que representa un intervalo de 6 horas, y tiene asignado un precio. A partir de esta clase, pude componer la clase Vector. Ésta representa una de las energías que se desean medir (de aquí que tenga como atributos un nombre identificativo y una unidad de medida), y tiene una colección de Tramos que se utilizará para calcular el precio del consumo en cada momento del día.

Por último, creé la clase Config. Ésta se puede entender como la aplicación del contrato energético vigente en un día concreto: tiene un atributo para comprobar su vigencia, así como la divisa en la que se contabiliza el consumo, y la colección de vectores energéticos de los cuales esa configuración puede ofrecer seguimiento.

De manera independiente a lo que constituiría las clases de datos propias de la aplicación, hice uso de otra clase, dedicada a implementar los métodos de serialización y deserialización necesarios para guardar y cargar archivos de configuración. Ésta es producto de la colaboración con el equipo de la empresa, y estaba destinada a ser una simple librería de métodos (de ahí que se declarase como clase estática).

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

```
using ...

namespace Configurar_Precios.Classes
{
    [DataContract]
    public class Tramo
    {
        public Tramo() { }

        public Tramo(int i, decimal p) ...

        [DataMember]
        public string Hora_Ini { get; set; }

        [DataMember]
        public string Hora_Fin { get; set; }

        [DataMember]
        public decimal Precio { get; set; }
    }
}

using ...

namespace Configurar_Precios.Classes
{
    [DataContract]
    public class Vector
    {
        public Vector(...)

        public Vector(Vector v) ...

        [DataMember]
        public string Nombre { get; set; }

        [DataMember]
        public string Unidad_Medida { get; set; }

        [DataMember]
        public List<Tramo> Tramos { get; set; }
    }
}

using ...

namespace Configurar_Precios.Classes
{
    [DataContract]
    public class Config
    {
        public Config(...)

        [DataMember]
        public string Nombre_Config { get; set; }

        [DataMember]
        public string Dir_CSV { get; set; }

        [DataMember]
        public List<Vector> Energias { get; set; }

        [DataMember]
        public string Vigencia { get; set; }

        [DataMember]
        public string Divisa { get; set; }
    }
}

using ...

namespace Configurar_Precios.Classes
{
    public static class Serializacion
    {
        public static string Serialize<T>(this T obj) ...

        public static void SerializeToFile<T>(this T obj, string pFileName) ...

        public static T Deserialize<T>(this string serialized) ...

        public static T Deserialize<T>(this string serialized, Type pType) ...

        public static T DeserializeFromFile<T>(string pFileName) ...

        public static T DeserializeFromFile<T>(string pFileName, Type pT) ...
    }
}
```

Ilustración 186: Clases de datos

A excepción de la clase Serialización, el resto conforman una jerarquía que es necesario guardar y cargar. Para ello decidí utilizar la librería Serialization (de .NET, distinta a la clase de datos implementada), que, mediante la asignación de atributos, tanto a la clase, como a las propiedades de ésta, permite una serialización conveniente a XML, y una deserialización que recupera en un objeto de la clase todas las propiedades.

Mediante el uso de `[DataContract]`, se indica a la librería que, llegado el momento de serializar, estamos hablando de una clase de la cual queremos una representación en XML. De igual manera, `[DataMember]` indica que si la clase a la que pertenece se va a serializar, la propiedad también lo hará.

Una limitación de este procedimiento es que no es posible serializar métodos. A primera vista puede parecer un gran inconveniente, pero no hay que olvidar que, cualquier código que pretenda reconstruir un objeto, a partir de XML, necesitará conocer la clase de dicho objeto. De esta manera, se crea una

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

instancia vacía de una clase, con todos sus procedimientos y propiedades (inicialmente vacías), y se rellena con los datos obtenidos del XML.

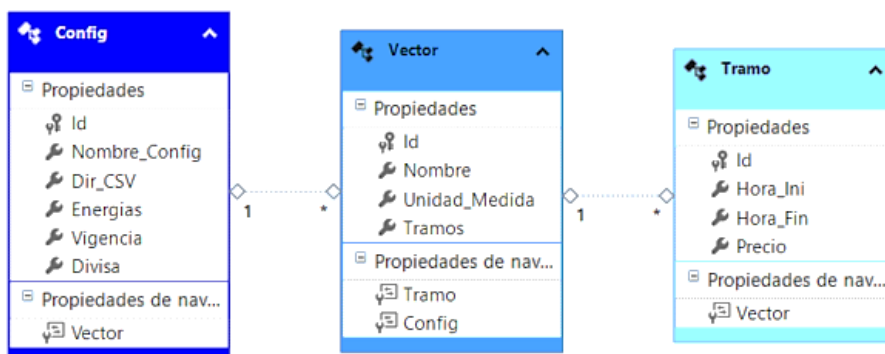


Ilustración 187: Diagrama de Clases

Empezando desde abajo: UserControl para Vector

Fiel a la filosofía bottom-up, lo primero es crear una manera de generar un nuevo vector.

La idea seguida para desarrollar este control se basa en la creación de una variable pública de tipo Vector, que servirá de contenedor temporal para los datos que queramos transferir a la configuración final. En el código propio de la ventana donde se sitúe el control, se creará un nuevo vector, y se igualará a la variable del mismo tipo en el control, de manera que se generará una referencia entre ambos. Una vez la referencia está creada, dentro del control se asignan valores a las propiedades del vector temporal, y gracias a la igualdad entre objetos, el vector creado en la ventana verá los cambios. Posteriormente se insertará ese mismo vector en la colección de Vectores de la Configuración en cuestión.

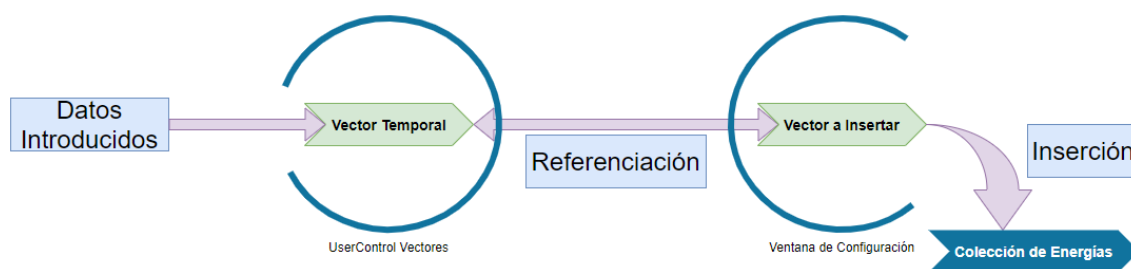


Ilustración 188: Flujo de datos desde UserControl

Cabe decir que, aunque la información insertada en la colección de energías pueda ser diferente cada vez, el vector utilizado es siempre el mismo, pero conteniendo datos distintos, debido a la referencia creada con la variable del control.

Hora Inicio	Hora Fin	Precio
00:00	04:00	
04:00	08:00	
08:00	12:00	
12:00	16:00	
16:00	20:00	
20:00	00:00	

Descartar Valores		Guardar Tramos	
Nombre del Vector			
Unidad de Medida			
Guardar Vector			

Ilustración 189: UserControl para Vector

Siguiendo la filosofía de simplificar el uso de la aplicación, el control cuenta con una pequeña tabla, que ilustra sobre qué intervalo horario se aplica el precio introducido, así como dos entradas de texto más para el nombre de la energía, y la unidad de medida utilizada. El botón “Descartar Valores” simplemente limpia las entradas de texto para los precios, mientras que “Guardar Tramos” hace que se genere una colección de tramos, que posteriormente se asignará al vector energético. Para hacer esto, crea una colección de objetos del tipo Tramo, a la que añade entradas nuevas, utilizando el constructor parametrizado de la misma clase. Posteriormente, borra los tramos del vector SelectedVector (es decir, el vector que manipulamos y usamos para la inserción en la colección de energías), y implementa un recorrido de la colección temporal, insertando cada uno de los tramos que encuentra en SelectedVector. De esta manera, nos aseguramos de que el vector insertado en la colección de energías tenga los tramos que el usuario haya definido en el control. Más allá de esta funcionalidad, el botón también controla que haya la información suficiente como para definir el vector, y vacía los campos para poder introducir nuevos datos.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

```
private void Button2_Click(object sender, EventArgs e)
{
    if(textBox1.Text == "" || textBox2.Text == "" || textBox3.Text == "" || textBox4.Text == "" || textBox7.Text == "" || textBox8.Text == "")
    {
        MessageBox.Show("Es necesario rellenar todos los campos", "Aviso");
    }
    else
    {
        if(SelectedVector == null)
        {
            MessageBox.Show("Es necesario que exista un vector", "Aviso");
        }
        else
        {
            List<Tramo> tempListTramo = new List<Tramo>();

            tempListTramo.Add(new Tramo(System.Convert.ToInt32(0), System.Convert.ToDecimal(textBox1.Text)));
            tempListTramo.Add(new Tramo(System.Convert.ToInt32(1), System.Convert.ToDecimal(textBox2.Text)));
            tempListTramo.Add(new Tramo(System.Convert.ToInt32(2), System.Convert.ToDecimal(textBox3.Text)));
            tempListTramo.Add(new Tramo(System.Convert.ToInt32(3), System.Convert.ToDecimal(textBox4.Text)));
            tempListTramo.Add(new Tramo(System.Convert.ToInt32(4), System.Convert.ToDecimal(textBox7.Text)));
            tempListTramo.Add(new Tramo(System.Convert.ToInt32(5), System.Convert.ToDecimal(textBox8.Text)));

            SelectedVector.Tramos.Clear();
            foreach (var t in tempListTramo)
            {
                SelectedVector.Tramos.Add(t);
            }
            if (infoAvailable < 2) infoAvailable++;
        }
        textBox1.Text = textBox2.Text = textBox3.Text = textBox4.Text = textBox7.Text = textBox8.Text = "";
    }
}
```

Ilustración 190: Guardado de Tramos

Más abajo, los dos campos para el nombre y la unidad de medida cuentan con su propio botón de guardado. Decidí separar el guardado de los tramos y el del nombre y la unidad de medida para posibilitar la inserción de varios vectores con los mismos tramos, o con los mismos nombres y unidad, solo por si resultaba útil en algún contexto. Para evitar entradas erróneas, las operaciones de guardado no se realizan a menos que todos los campos de los que dependen tengan datos. De igual manera, al realizar un guardado correcto, se vacía el campo de texto para evitar duplicados.

Editando la configuración: Form de atributos

Después de crear una manera de editar un vector, hay que diseñar una manera para acabar de editar nuestra configuración.

En tanto que una aplicación de Windows Forms, la manera más conveniente de editar configuraciones es desde Forms. Por esta razón, se creó un Form nuevo, con el UserControl para vectores incrustado.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Atributos de la Configuración

Hora Inicio

00:00

Hora Fin

04:00

Precio

04:00

08:00

08:00

12:00

12:00

16:00

16:00

20:00

20:00

00:00

Descartar Valores

Guardar Tramos

Nombre del Vector

Unidad de Medida

Guardar Vector

Nombre del Vector

Unidad

Precio 0

Precio 1

Precio 2

Precio 3

Precio 4

Precio 5

Añadir Vector a Fuentes

Borrar Fuentes

Nombre:

Definir Nombre

Directorio de Guardado:

Definir Directorio

Fecha de Vigencia:

lunes 15 de julio de 2019

Divisa:

Definir Divisa

Serializar Vectores a CSV

Serializar Configuración a XML

Ilustración 191: Edición de configuraciones

Al cargar la ventana, lo primero que hace es crear un nuevo Vector energético, e igualarlo a la variable de tipo Vector del UserControl. Ahora, ya hay una estructura de datos sobre la que añadir la información que el usuario introduzca. Adicionalmente, se inicializa de manera distinta la nueva configuración, dependiendo de si en el menú principal se escogió empezar de cero, o cargar una configuración existente para trabajar sobre ella.

```
private void Form_Atributos_Config_Load(object sender, EventArgs e)
{
    if (tipo == 0)
    {
        tempV = new Vector();
        ucVector1.SelectedVector = tempV;

        tempC = new Config();
    }
    else if (tipo == 1)
    {
        tempV = new Vector();
        ucVector1.SelectedVector = tempV;

        tempC = Serializacion.DeserializeFromFile<Config>(pathXML);
        Cargar_LV();
    }
}
```

Ilustración 192: Creación del nuevo Vector

A partir de aquí, el usuario podrá visualizar, en la lista de la ventana, los vectores energéticos que haya añadido a la configuración actual, así como eliminarlos,

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

si de cometió algún error. El resto de campos simplemente se ocupan de rellenar las propiedades de la clase Config, y así hasta llegar a los dos botones finales.

```
private void button1_Click_1(object sender, EventArgs e)
{
    var records = new List<Tramo>();
    foreach (var v in tempC.Energias)
    {
        records = v.Tramos;

        string filenameCSV = System.IO.Path.Combine(tempC.Directorío, dtp.Value.ToString("dd_MM_yyyy") + "_" +
            tempC.Nombre_Config + "_" +
            v.Nombre + "_" +
            tempC.Divisa +
            ".csv");
        using (var writer = new System.IO.StreamWriter(filenameCSV))
        using (var csv = new CsvWriter(writer))
        {
            csv.WriteRecords(records);
        }
        File.WriteAllText(filenameCSV, File.ReadAllText(filenameCSV, Encoding.Default), Encoding.UTF8);
    }
}
```

Ilustración 193: Guardado de vectores a CSV

En el primero, se genera un fichero CSV por cada vector energético presente en la configuración, y dentro de cada fichero se especifican los tramos y sus correspondientes precios. En el nombre de cada fichero, se indica la fecha de vigencia de la configuración, el nombre de esta, el nombre del vector en cuestión, y la divisa en la que están representados los precios. Para realizar esta tarea, se utiliza CSVHelper, una librería externa importada a través de NuGet Packet Manager, el gestor de Visual Studio para la inclusión de nuevas librerías. Se implementa un recorrido a través de la colección de energías, y por cada uno de los vectores energéticos, se genera una ruta de archivo, que será donde se ubicará el fichero generado. Posteriormente se utiliza la librería externa para escribir las listas de tramos que representan cada vector en la ruta especificada.

El último botón se encarga de guardar toda la configuración en un fichero XML, con el objetivo de poder cargarlo posteriormente, en caso de necesitarlo. De igual manera, en el nombre del fichero figuran tanto la fecha de vigencia, como el nombre de la configuración en cuestión.

Escoger qué hacer: Form de menú principal

Esta ventana hace las veces de punto de entrada a la aplicación. Para ello, debe de ser clara en cuanto a las posibilidades que ofrece.

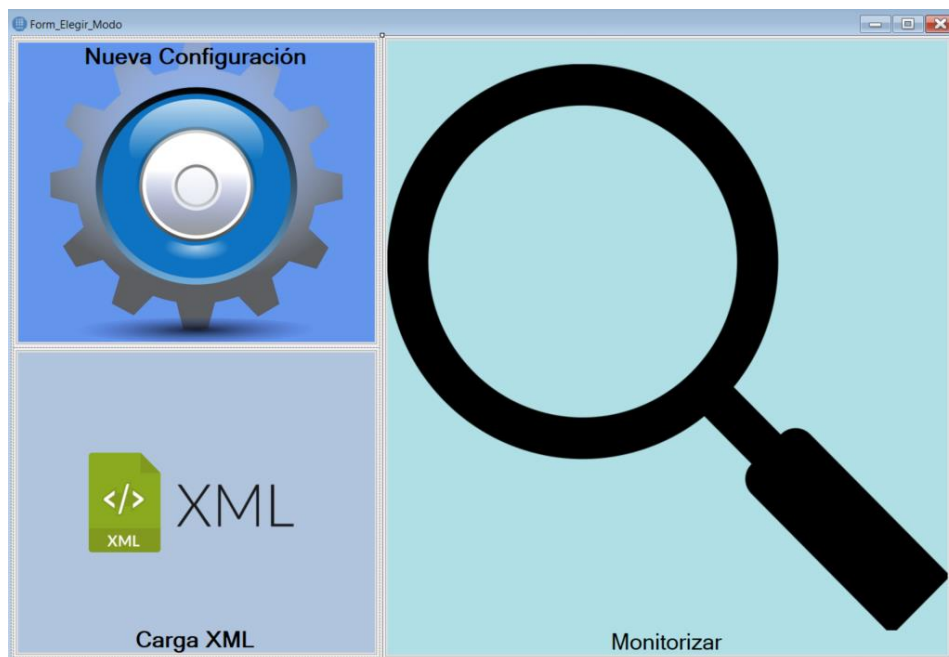


Ilustración 194: Form de elección de modo

Se diseñó un menú simple, de dimensiones manejables, que sirviese para acceder a las distintas funcionalidades del aplicativo. El botón con el engranaje permite acceder a la edición de configuraciones con una nueva configuración, mientras que la opción de abajo permite cargar una existente y trabajar sobre ella (útil si las variaciones no son muy grandes). En el caso de cargar XML, se hace uso de la clase Serialización, para reconstruir el objeto Config y tener de nuevo una instancia de la clase que se pueda manipular.

El botón más grande da acceso a la ventana de monitorización. En ella, se escoge una configuración vigente en el momento de ejecutar la aplicación, y se inicia la comunicación OPC con el servidor residente en la máquina. En el siguiente apartado de esta memoria se discute más en profundidad esta ventana.

El motivo de presentar el menú, antes que la ventana de monitorización, es que, en cuanto a código, la ventana de edición de configuraciones necesita saber del menú cómo queremos trabajar la configuración (de cero o cargada), pero para monitorizar no necesitamos ningún parámetro proveniente del menú, solo escoger una configuración ya existente para cargarla.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

```
private void Cargar_LV()
{
    listView1.Items.Clear();
    foreach (Vector v in tempC.Energias)
    {
        var row = new string[] {
            v.Nombre,
            v.Unidad_Medida,
            v.Tramos[0].Precio.ToString(),
            v.Tramos[1].Precio.ToString(),
            v.Tramos[2].Precio.ToString(),
            v.Tramos[3].Precio.ToString(),
            v.Tramos[4].Precio.ToString(),
            v.Tramos[5].Precio.ToString()
        };
        var lvi = new ListViewItem(row);
        lvi.Tag = v;
        listView1.Items.Add(lvi);
    }
}
```

Ilustración 195: Uso de Tag en el ListView

Para implementar la carga de XML, y poder utilizar la ventana de configuración de atributos tanto para nuevas configuraciones como para configuraciones cargadas por XML, tuve que implementar dos maneras distintas de cargar esta ventana, como ya se ha mencionado. El método `Cargar_LV` se encarga de hacer que la `ListView` muestre los vectores de la configuración cargada. Para ello, utiliza el atributo `Tag` que facilita C#.

Este atributo es un objeto asociado al control, de manera que, accediendo al control, se pueden tomar decisiones que afectan, o tienen que ver, con una instancia en concreto de una clase. Para explicar mejor este concepto, se puede observar como implementé el borrado de vectores de la colección vía el borrado de entradas en la `listView`.

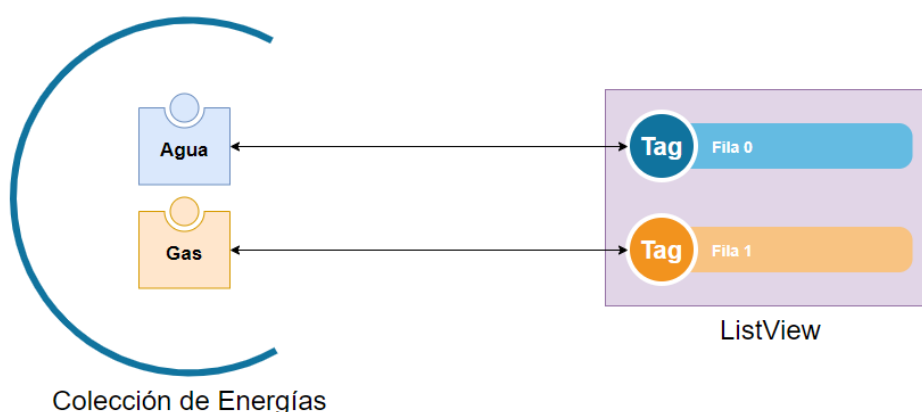


Ilustración 196: Vinculación mediante Tags

Aunque es cierto que cada entrada de la `listView` representa un vector en la colección de energías, no es cierto que ambos estén vinculados. Es decir, borrar la fila de la `listView` no borra el vector en la colección. Hay técnicas de Data Binding que posibilitan hacer esto, pero dado que no están por defecto en los controles, requerían un tiempo de implementación, y consideré que el uso de

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Tags era suficiente para el propósito. Es por eso que, tanto al añadir nuevos vectores como al cargar configuraciones, a la hora de construir la fila del listView, le asigno como Tag el vector del cual estoy extrayendo la información. De esta manera, en la función de borrado, antes de eliminar la fila, gracias al Tag puedo eliminar la instancia concreta de la clase Vector, y posteriormente eliminar su representación en el listView.

```
private void Button2_Click(object sender, EventArgs e)
{
    if(ucVector1.SelectedVector == null || ucVector1.infoAvailable != 2)
    {
        MessageBox.Show("No hay datos con los que generar el nuevo vector.", "Aviso");
    }
    else
    {
        tempC.Energias.Add(new Vector(ucVector1.SelectedVector));

        var row = new string[] {
            ucVector1.SelectedVector.Nombre,
            ucVector1.SelectedVector.Unidad_Medida,
            ucVector1.SelectedVector.Tramos[0].Precio.ToString(),
            ucVector1.SelectedVector.Tramos[1].Precio.ToString(),
            ucVector1.SelectedVector.Tramos[2].Precio.ToString(),
            ucVector1.SelectedVector.Tramos[3].Precio.ToString(),
            ucVector1.SelectedVector.Tramos[4].Precio.ToString(),
            ucVector1.SelectedVector.Tramos[5].Precio.ToString()
        };
        var lvi = new ListViewItem(row);
        lvi.Tag = tempC.Energias.Last();
        listView1.Items.Add(lvi);
    }
}
```

Ilustración 197: Añadido de filas y vinculación de Tags

Monitorización: ventana de consumos

La aplicación cobra sentido una vez podemos utilizar una configuración para calcular costes. En esta ventana lo hacemos en tiempo real.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

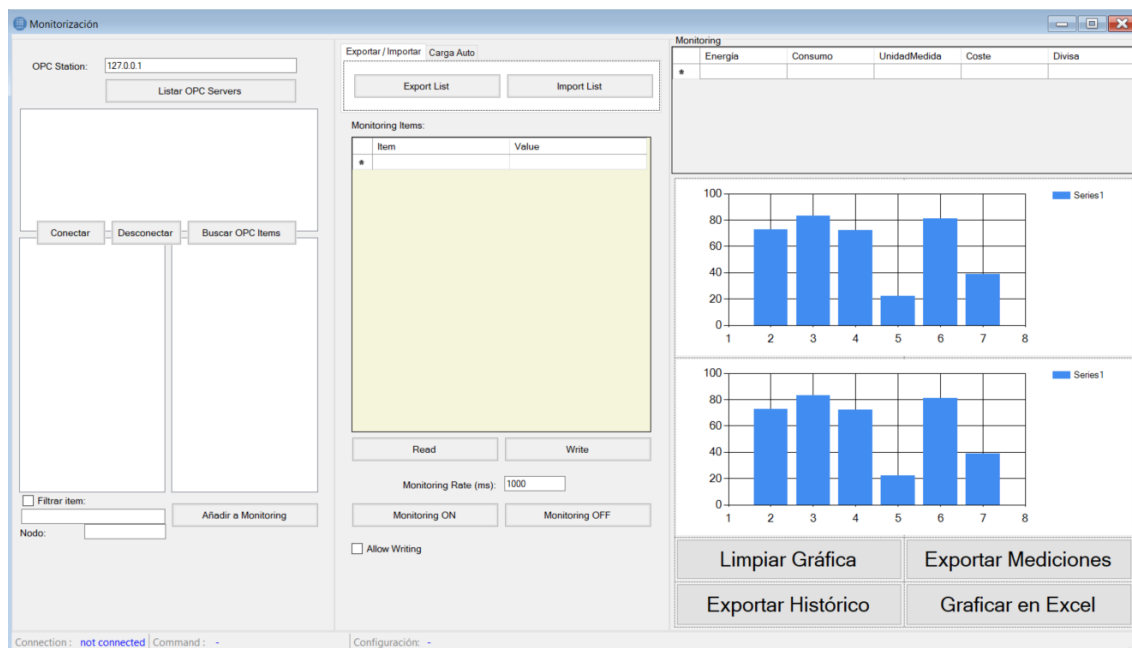


Ilustración 198: Pantalla de monitorización

Después de cargar una configuración válida, aparece la ventana de monitorización. El primer paso a seguir sería listar los servidores OPC disponibles en la máquina local. Esta aplicación limita la visibilidad de servidores OPC a la máquina actual. Los otros tipos de OPC, que ofrecían más visibilidad, apenas tenían documentación sobre cómo hacerlos interactuar con .NET, y además muchos utilizaban software de pago, que no se contemplaba para el proyecto.

Una vez realizamos la conexión con el servidor deseado, se nos muestra, en forma de árbol, los canales que éste maneja, y, en un nivel más bajo de la jerarquía, podemos escoger el dispositivo. Al hacerlo, se muestra los tag (señales) que están disponibles, y, seleccionándolas, se añaden al cuadro de monitorización.

```
public struct strOPCServer
{
    public OPCServer MyServer;
    public OPCGroup MyGroup;
    public OPCItem MyItem;
    public Array ServerHandles;
    public Array Errors;
    public Array Values;
    public int MonCount;
}

public struct strcMonItem
{
    public int _Server;
    public string _Name;
    public object _Value;
    public int _Index;
    public object _ServerHandles;
}
```

Ilustración 199: Declaración de las estructuras necesarias

A partir de aquí, podemos escoger si realizar lecturas de manera manual, o activar el modo Monitoring mediante los botones del panel central, para recibir nuevos valores cada tiempo de refresco. Las lecturas actualizan tanto la tabla como la gráfica del panel derecho. Los botones bajo la gráfica permiten, por un lado, limpiar la gráfica en caso de que se haya vuelto incómoda de leer, exportar a CSV los consumos actuales y el histórico de lecturas hasta el momento, generar un fichero por cada vector monitorizado, que contiene el histórico de lecturas desde que se inició la aplicación, y finalmente generar una libro de cálculo con tantas hojas como vectores monitorizados, que muestra el histórico de consumos en cada instante, y lo grafica para poder detectar picos o tendencias.

En el código, esta ventana también incluye validaciones sobre la fecha de vigencia de la configuración escogida, y también comprueba, antes de leer valores, que la fecha no haya cambiado, entre otras comprobaciones.

```
dataCoste.Rows[i - 1].Cells["Coste"].Value = Convert.ToDecimal(dataCoste.Rows[i - 1].Cells["Coste"].Value) +  
    ((Convert.ToDecimal(ItemValues.GetValue(i)) - Convert.ToDecimal(dataCoste.Rows[i - 1].Cells["Consumo"].Value)) * config.Energias[i - 1].Tramos[0].Precio);  
dataCoste.Rows[i - 1].Cells["Consumo"].Value = ItemValues.GetValue(i);  
chart.Series[i - 1].Points.AddXY(DateTime.Now.ToLongTimeString(), Convert.ToDouble(Convert.ToDecimal(ItemValues.GetValue(i)) * config.Energias[i - 1].Tramos[0].Precio));  
chart.Series[i - 1].Points.Last().Tag = DateTime.Now;
```

Ilustración 200: Añadido de puntos/lecturas

Cada vez que se obtiene un nuevo dato, se calcula el precio acumulado, teniendo en cuenta cuánto ha crecido el consumo desde la última lectura, se añade un punto a la gráfica, y también se le asigna, como Tag, una marca de tiempo. Ésta última nos permitirá crear un histórico de lecturas, que luego podremos exportar.

OPCAutomation: puesta en marcha del cliente OPC

Siguiendo la documentación de la librería externa. Se inicializan las variables que permitirán crear el cliente OPC.

Lo primero que hay que hacer es crear una nueva instancia del tipo OPCServer. Posteriormente se realiza la conexión mediante el método Connect ofrecido por la clase OPCServer. En mi caso, utilicé el descubrimiento de servidores, y el texto mostrado en los controles de la ventana, para darle al procedimiento los argumentos que necesitaba.

```
MyOPCServer[iOPC].MyServer.Connect(1stOPCServer.SelectedNode.Text, txtNode.Text);
```

Ilustración 201: Conexión al servidor OPC

Lo siguiente fue crear un grupo. El grupo es la entidad que engloba los ítems, es decir, las señales que trataremos, y en parte define cómo éstas se comportan. Por lo tanto, no es posible crear un cliente funcional y correcto en forma si no se

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

sigue esta arquitectura de servidor-grupo-item. Más allá de esto, hay que configurar manualmente ciertas características del grupo. En este caso, se definió que no se deseaba activar el grupo nada más crearlo, pero, por el contrario, sí que era necesario suscribirse (hay que recordar que OPC funciona por suscripciones: el servidor surte al cliente los datos a los que está suscrito). También se define la frecuencia con la que el cliente obtendrá nuevos datos, mediante una entrada de texto colocada para tal propósito. Por último, se definen los handlers (funciones) que se ejecutan cuando hay un cambio de datos, y cuando se realizan tanto lecturas como escrituras asíncronas.

```
MyOPCGroup[iOPC].IsActive = false;  
MyOPCGroup[iOPC].IsSubscribed = true;  
MyOPCGroup[iOPC].UpdateRate = Convert.ToInt32(txtRate.Text);  
MyOPCGroup[iOPC].DataChange += new DIOPCGroupEvent_DataChangeEventHandler(MyOPCGroup_DataChange);  
MyOPCGroup[iOPC].AsyncWriteComplete += new DIOPCGroupEvent_AsyncWriteCompleteEventHandler(MyOPCGroup_AsyncWriteComplete);  
MyOPCGroup[iOPC].AsyncReadComplete += new DIOPCGroupEvent_AsyncReadCompleteEventHandler(MyOPCGroup_AsyncReadComplete);
```

Ilustración 202: Inicialización del grupo OPC

Una vez definido el grupo, se puede pasar a crear ítems, cuyo valor se utilizará para monitorizar el consumo y calcular los costes. Para ello se añade una nueva fila en la lista de ítems monitorizados, y se añade una posición a la colección de los mismos. Después se definen los atributos del último ítem de la colección (que es el último añadido). Por último, se añade a los ítems asociados al grupo OPC, y se define el ServerHandle, que es un número utilizado por el servidor OPC para llevar a cabo lecturas y escrituras sobre un determinado ítem.

```
dataOPC.Rows.Add();  
dataOPC[0, MonItem_Count].Value = itemName;  
  
MonItem_Count += 1;  
  
Array.Resize(ref MonItem, MonItem_Count + 1);  
MonItem[MonItem_Count]._Server = iOPC;  
MonItem[MonItem_Count]._Name = itemName;  
MonItem[MonItem_Count]._Index = dataOPC.RowCount - 1;  
  
MyOPCServer[iOPC].MyItem = MyOPCGroup[iOPC].OPCItems.AddItem(itemName, MonItem_Count);  
MyOPCServer[iOPC].ServerHandles.SetValue(MyOPCServer[iOPC].MyItem.ServerHandle, MonItem_Count);
```

Ilustración 203: Añadido de un ítem OPC

Para adquirir datos de manera asíncrona, se utiliza el botón “Read”, que suscribe los ítems que se quiere monitorizar, y ejecuta el procedimiento AsyncRead del grupo OPC. Al completarse la lectura, se dispara de manera automática el evento MyOPCGroup_AsyncWriteComplete, y con él las instrucciones para actualizar los consumos, los costes y la gráfica.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

```
MyOPCGroup[iOPC].AsyncRead(numItems, MyOPCServer[iOPC].ServerHandles, out MyOPCServer[iOPC].Errors, DateTime.Now.Second, out transID);  
private void MyOPCGroup_AsyncReadComplete(int TransactionID, int NumItems, ref Array ClientHandles,  
ref Array ItemValues, ref Array Qualities, ref Array TimeStamps, ref Array Errors)...
```

Ilustración 204: Método y evento de lectura asíncrona

Por el contrario, para adquirir datos de manera síncrona, se tiene que usar los botones de activación/desactivación de la modificación. Estos determinan si el grupo está activo y suscrito, y activan el botón que realiza la función contraria, para poder detener la llegada automática de datos cuando se desee. La propia librería implementa un evento que se dispara automáticamente al llegar datos en tiempo real, y al capturarlo con el handler `MyOPCGroup_DataChange`, se pueden tener actualizaciones en la tabla de datos y en la gráfica en tiempo real.

```
if (MonItem_Count == 0)  
{  
    SubscribeItems();  
}  
  
for (int i = 0; i <= 10; i++)  
{  
    if (MyOPCServer[i].MyServer.ServerState == 1)  
    {  
        MyOPCGroup[i].IsActive = true;  
    }  
}  
btnMonitoringOn.Enabled = false;  
btnMonitoringOff.Enabled = true;  
  
private void MyOPCGroup_DataChange(int TransactionID, int NumItems, ref Array ClientHandles,  
ref Array ItemValues, ref Array Qualities, ref Array TimeStamps)...
```

Ilustración 205: Monitorización activada y evento síncrono

Por último, cuando se va a cerrar la ventana de monitorización, se dispara el evento `FormClosing`, y se ejecuta un handler que simula un click en el botón de desconexión. Este último evento, a su vez, realiza toda una serie de acciones para borrar los grupos OPC creados, y sacar los ítems monitorizados del estado activo, cosa que no sucede de manera automática, y desconectarse efectivamente de los servidores.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

```
for (int i = 0; i <= 10; i++)
{
    MyOPCServer[i].MyItem = null;
    if (MyOPCServer[i].MyServer.ServerState == 1)
    {
        MyOPCServer[i].MyServer.OPCGroups.RemoveAll();
        MyOPCGroup[i].IsSubscribed = false;
        MyOPCGroup[i].IsActive = false;
    }
    MyOPCGroup[i] = null;
}

bool okDiskonek = true;
for (int i = 0; i <= 10; i++)
{
    if (MyOPCServer[i].MyServer.ServerState == 1)
    {
        MyOPCServer[i].MyServer.Disconnect();
        if (MyOPCServer[i].MyServer.ServerState == 1)
        {
            okDiskonek = false;
        }
    }
}
```

Ilustración 206: Borrado de grupos y desconexión de servidores

EPPlus: creación de la hoja de cálculo

El uso de esta librería permite crear hojas de cálculo programáticamente.

Para agilizar el proceso de Mejora Continua, es útil disponer de manera directa de una fuente intuitiva de información como es una gráfica. Es cierto que se ofrece la opción de exportar a formato CSV los valores leídos, pero, aunque sea una manera conveniente de llevar datos de una aplicación a otra, no resulta eficiente, de cara a un proceso de análisis de datos.

```
using (var p = new ExcelPackage())
{
    for (int i = 0; i < chart.Series.Count; i++)
    {
        var ws = p.Workbook.Worksheets.Add(chart.Series[i].Name);
        for (int j = 0; j < chart.Series[i].Points.Count; j++)
        {
            ws.Cells["A" + (j + 1).ToString()].Value = Convert.ToDateTime(chart.Series[i].Points[j].Tag).ToLongDateString();
            ws.Cells["B" + (j + 1).ToString()].Value = chart.Series[i].Points[j].YValues[0];
        }

        var ct = ws.Drawings.AddChart("Lecturas", OfficeOpenXml.Drawing.Chart.eChartType.LineMarkers);
        ct.SetPosition(0, 0, 2, 0);
        ct.SetSize(1000, 1000);
        ct.RoundedCorners = true;
        var serieConsumo = ct.Series.Add("B1:B" + (chart.Series[0].Points.Count).ToString(), "A1:A" + (chart.Series[0].Points.Count).ToString());
        serieConsumo.Fill.Color = Color.Navy;
    }

    if (folderBrowserDialog2.ShowDialog() == DialogResult.OK)
    {
        p.SaveAs(new FileInfo(System.IO.Path.Combine(folderBrowserDialog2.SelectedPath, "Readings_" + config.Nombre_Config +
            "_" + DateTime.Now.ToLongDateString() + "_" + ".xlsx")));
    }
}
```

Ilustración 207: Generación de la hoja de cálculo

Primeramente, se crea una instancia de ExcelPackage. Es el objeto que permite generar hojas de cálculo, y editarlas para conseguir algo que cumpla nuestras necesidades. En este caso, se crea una hoja de cálculo por cada vector monitorizado y, dentro de ésta, se muestra la gráfica, cuya serie de datos se forma a través de los valores escritos programáticamente en las celdas. Finalmente se guarda el libro en la ubicación que el usuario escoja, con un

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

nombre que identifique la configuración utilizada, y el timestamp de cuándo se solicitó la generación de dicho libro.

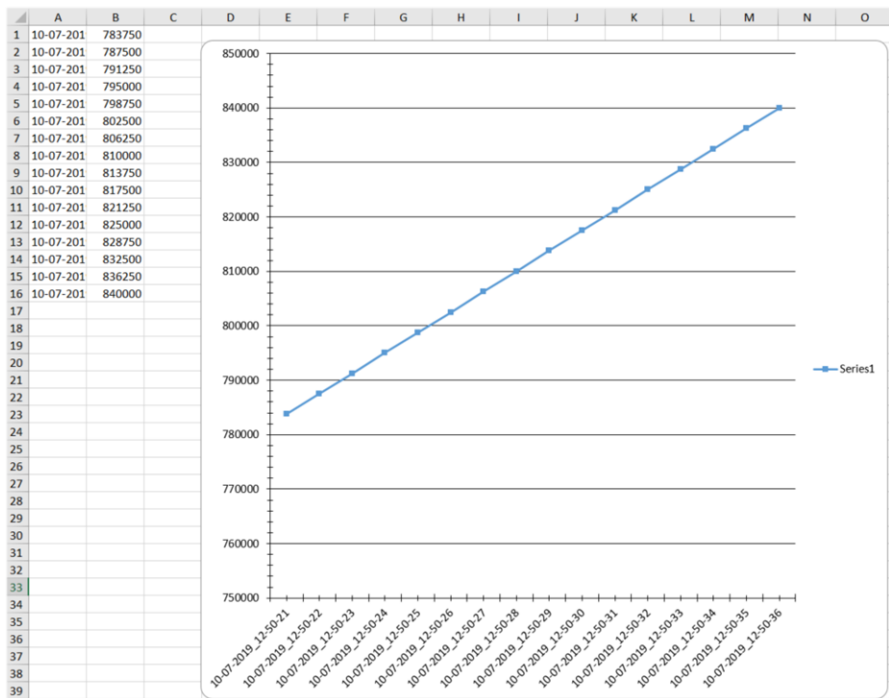


Ilustración 208: Hoja de cálculo generada

Testeo: Comunicación Modbus con Microcontrolador

Para comprobar que el sistema, como integración de tecnologías, funciona, se usa una placa Arduino.

Circuito Digital

A efectos teóricos, utilizar simulaciones es útil, en tanto que permite demostrar que la solución debería de funcionar, en un hipotético despliegue real. Para romper las limitaciones que existen al trabajar con abstracciones, se construyen dos circuitos con una placa Arduino: uno destinado a la lectura y escritura de señales digitales, y otro a leer y escribir señales analógicas.

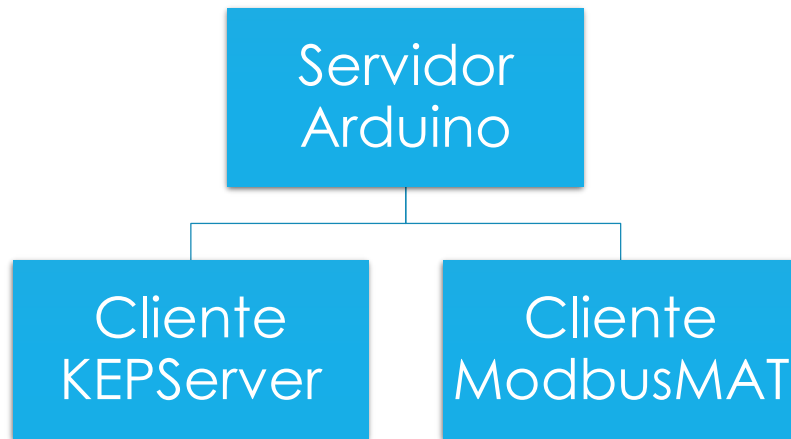


Ilustración 209: Arquitectura de Testeo

A priori, parece un problema sencillo de programación en Arduino, pero aquí, ésta hace las veces de servidor Modbus, y realizará acciones mediante comprobaciones contra sus registros de datos. Para cotejar las lecturas, en paralelo, se tendrán tanto KEPServer como ModbusMAT, un freeware que permite inspeccionar los valores de cada dirección del servidor Modbus.

Para hacer posible montar el servidor Modbus en el Arduino, se hizo uso de la librería ArduinoModbus, oficial del fabricante. La idea para el circuito analógico era configurar dos coil (registros booleanos). Uno diría si se ha activado un botón, y el otro dictaría si hay que apagar un LED, que por defecto está encendido. Extrapolado a la industria, esta lógica es idéntica a la de un botón destinado a realizar paradas de emergencia: si se aprieta el botón, la máquina para. Como comentario, para efectos de los testeos, se escribió el código de tal manera que el LED se enciende si se deja de pulsar el botón. Ésta característica permite ver las fluctuaciones en la señal analógica a través de las gráficas de monitorización.

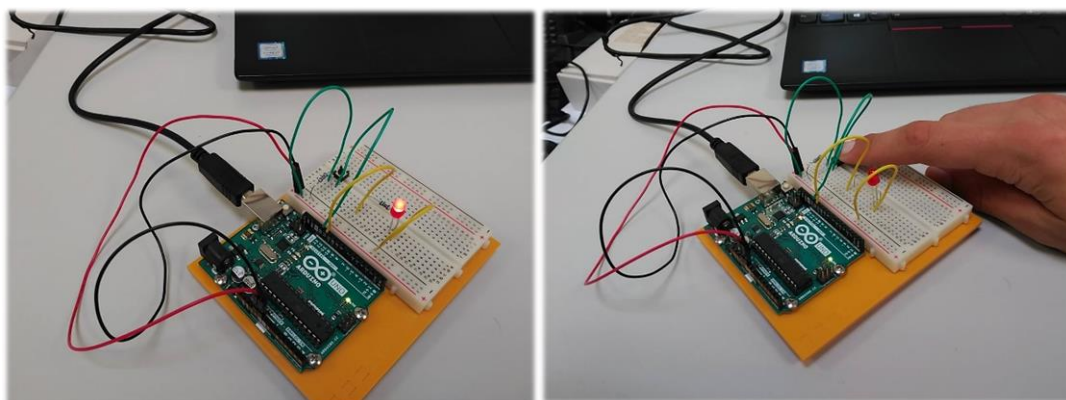


Ilustración 210: Circuito Digital

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Para el circuito digital, se usaron los pines digitales del Arduino, así como las salidas de 5V y la conexión a tierra (GND). Además de eso, se usó un LED, para comprobar que la escritura digital se realizaba, y dos resistencias, para evitar cortocircuitos.

```
#include <ArduinoRS485.h>
#include <ArduinoModbus.h>

void setup() {
    // put your setup code here, to run once:
    pinMode(13, INPUT_PULLUP);
    pinMode(12, OUTPUT);
    digitalWrite(12, LOW);

    ModbusRTUServer.begin(1, 9600);

    ModbusRTUServer.configureCoils(0,2);
    ModbusRTUServer.coilWrite(0, 0);
    ModbusRTUServer.coilWrite(1, 1);
}

void loop() {
    // put your main code here, to run repeatedly:
    if(digitalRead(13) == LOW){
        ModbusRTUServer.coilWrite(0, 0);
        ModbusRTUServer.coilWrite(1, 1);
    }
    else {
        ModbusRTUServer.coilWrite(0, 1);
        ModbusRTUServer.coilWrite(1, 0);
    }

    if(ModbusRTUServer.coilRead(1) == 1){
        digitalWrite(12, HIGH);
    }
    else{
        digitalWrite(12, LOW);
    }

    ModbusRTUServer.poll();
    delay(500);
}
```

Ilustración 211: Modbus Digital

A la hora de implementar el servidor, se utilizó comunicación Serial vía el puerto USB del laptop de desarrollo. En la función setup(), se configura el pin de entrada (para detectar el pulsado del botón) y el de salida (para dar corriente al LED). Después, se inicializa el servidor, con un ID igual a 1, y una tasa de transmisión de 9600 baudios. Después de esto, se configuran 2 coils, que son los que se necesitan para la prueba, y se inicializan los coil con los valores por defecto adecuados.

En el bucle de funcionamiento loop(), se comprueba si el botón se ha pulsado, se actualizan los coils según los resultados, y posteriormente se comprueba el estado del coil de control del LED, para decidir si éste se enciende o se apaga.

Para comprobar que, efectivamente, los valores guardados en el servidor Modbus Arduino se corresponden con la realidad del circuito, se carga el programa en la placa, y se ejecuta ModbusMAT en el laptop, para poder comprobar en tiempo real los coils configurados para este circuito concreto.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Suponiendo que el coil 1 dice si llega corriente a través del pulsador, y el coil 2 si se suministra corriente el LED, se puede comprobar, gracias a la interfaz gráfica, que el funcionamiento interno de la placa ilustra fielmente lo que está sucediendo.

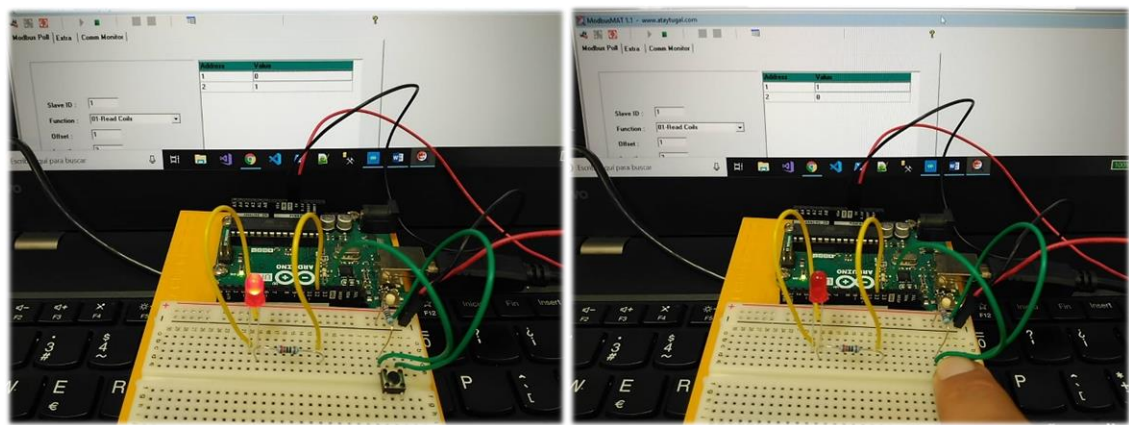


Ilustración 212: Comprobación de Coils en ModbusMAT

Como comprobación final, se ejecuta el programa de monitorización desarrollado. Este software, aunque pensado para calcular los costes derivados de un consumo incremental, y teniendo en cuenta una configuración preestablecida, nos servirá para visualizar, en tiempo real, los cambios en las señales (y servirá como justificación final de la utilidad del programa). Por lo tanto, se elabora una configuración donde el precio de cada tramo horario es 0, ya que solo nos interesa ver como fluctúan las señales.

[-] Identification	
Name	DigitalInput
Description	
[-] Data Properties	
Address	000001
Data Type	Boolean
Client Access	Read/Write
Scan Rate (ms)	100
[-] Identification	
Name	DigitalOutput
Description	
[-] Data Properties	
Address	000002
Data Type	Boolean
Client Access	Read/Write
Scan Rate (ms)	100

Ilustración 213: Mapeo Digital

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

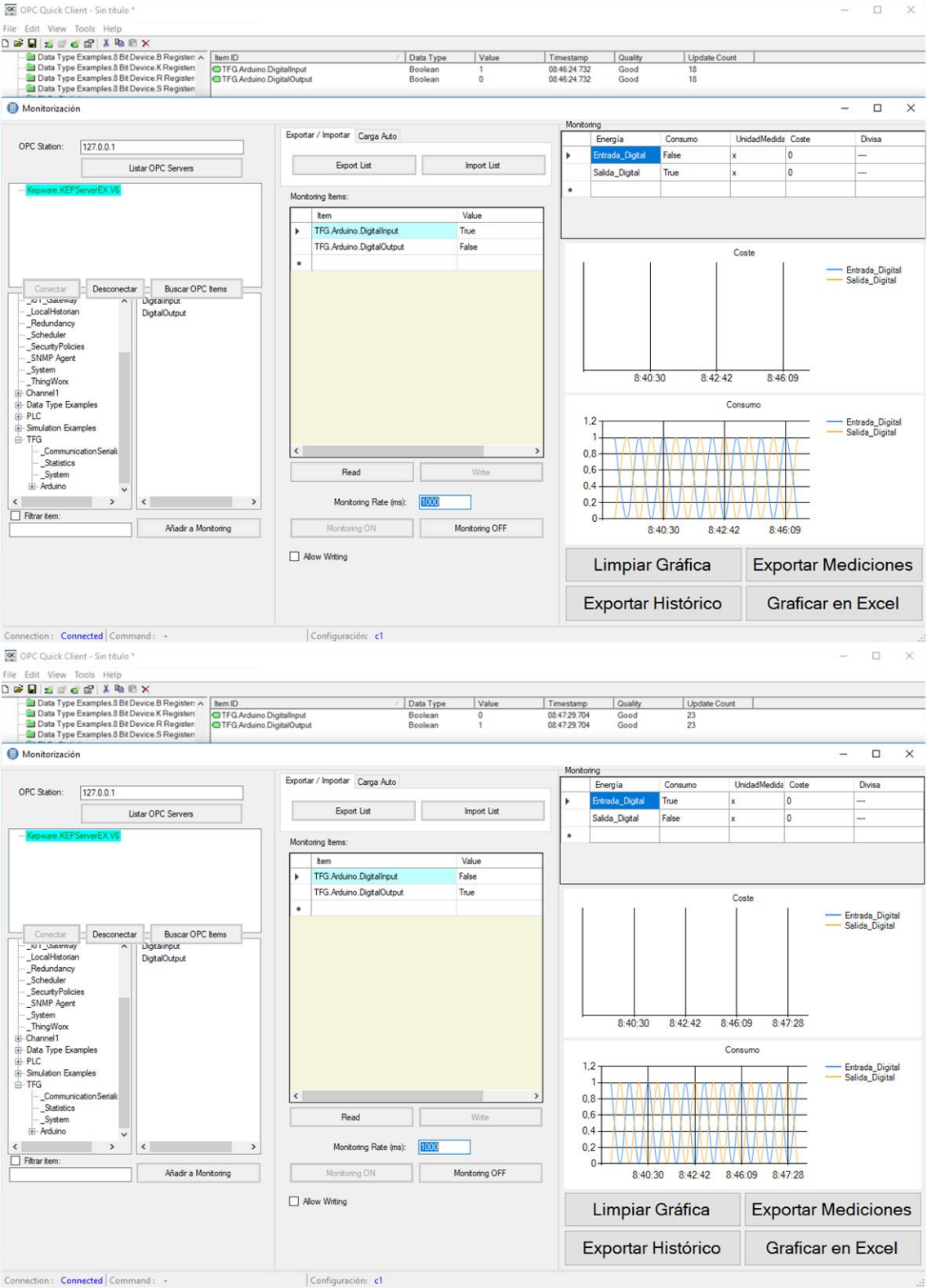


Ilustración 214: Monitorización del Circuito Digital

Después de comprobarlo con el software desarrollado, se aprecia cómo se dibuja una señal sinusoidal que varía cada vez que se aprieta el pulsador, y

cómo los valores de los tags monitorizados (correspondientes a los coils configurados) cambian según lo previsto.

Circuito Analógico

En industria, no todo es binario. En muchos casos, existen todo un rango de valores intermedios entre el estado de apagado y el estado de carga máxima. A efectos de un PLC que maneje Modbus, esto se traduce a una señal analógica que puede, por ejemplo, venir de un potenciómetro de algún tipo, y una respuesta, también analógica, que el propio autómatas genera, teniendo en cuenta la señal recibida.

Para hacer una demostración del desempeño de la aplicación de monitorización en entornos analógicos, se diseñó un pequeño circuito, que cuenta con un potenciómetro y un LED, además de resistores. Como trata con señales analógicas, el circuito utiliza los pines “analógicos” del Arduino, que tienen capacidades PWM.

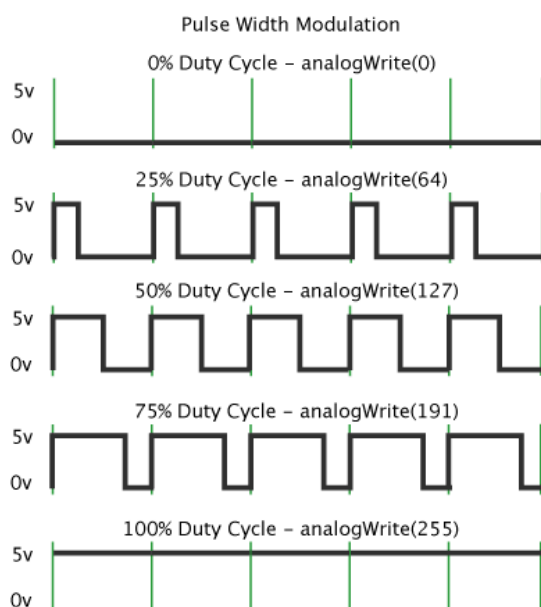


Ilustración 215: Pulse Width Modulation

El concepto de Pulse Width Modulation hace referencia a la adquisición de señales analógicas por medios digitales. Para conseguir esto, se utiliza el concepto de Duty Cycle, que es, básicamente, cuánto tiempo, dentro de un periodo, se suministran 5V. Gracias a poder determinar el tiempo que una señal está activa, se pueden simular voltajes dentro del rango de 0V a 5V, consiguiendo así los efectos que un voltaje real de ese tipo causaría. Sabiendo esto, y haciendo uso de las librerías de Arduino, podemos encender un LED con un voltaje que depende de la corriente que llega de un potenciómetro.

La lógica seguida en el código es la misma que para el circuito digital, salvo un par de detalles: en este caso no podremos usar Coils para almacenar la información, y además tendremos que mapear las lecturas del potenciómetro.

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

```
#include <ArduinoRS485.h>
#include <ArduinoModbus.h>

const int analogInPin = A0;
const int analogOutPin = 9;

int sensorValue = 0;
int outputValue = 0;

void setup() {
  ModbusRTUServer.begin(1, 9600);
  ModbusRTUServer.configureHoldingRegisters(0, 2);
  ModbusRTUServer.holdingRegisterWrite(0, 0);
  ModbusRTUServer.holdingRegisterWrite(1, 0);
}

void loop() {
  sensorValue = analogRead(analogInPin);
  ModbusRTUServer.holdingRegisterWrite(0, sensorValue);

  outputValue = map(ModbusRTUServer.holdingRegisterRead(0), 0, 1023, 0, 255);
  ModbusRTUServer.holdingRegisterWrite(1, outputValue);
  analogWrite(analogOutPin, ModbusRTUServer.holdingRegisterRead(1));

  ModbusRTUServer.poll();
  delay(2000);
}
```

Ilustración 216: Modbus Analógico

Al estar tratando con valores discretos, tenemos que utilizar registros capaces de almacenar valores que ocupen más bytes que un booleano: éstos son los Holding Registers. Para las necesidades del testeo, dos registros son suficientes, y se configuran igual que los Coils: especificando una dirección inicial, y cuántos registros se necesitan. En cuanto al mapeo de las lecturas, tiene que ver con que el sensor da valores entre 0 y 1023, mientras que la función de escritura analógica, utilizada para encender el LED, acepta parámetros de 0 a 255. Por lo tanto, se tiene que mapear los valores del potenciómetro a valores que la función de escritura pueda aceptar.

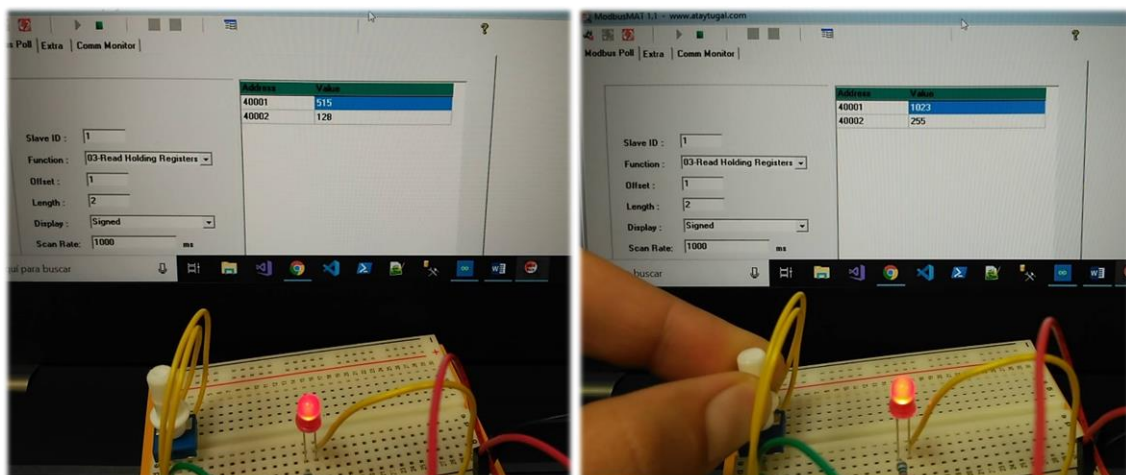


Ilustración 217: Luminosidad del LED

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Como en el anterior caso, se utiliza ModbusMAT para comprobar que los Holding Registers se actualizan adecuadamente, y reflejan la realidad del circuito. Es interesante observar el mapeo que se realiza al aplicar voltajes entre 0V y 5V: 0 en el potenciómetro significa 0 en el LED, 1023 en el potenciómetro significa 255 en el LED, pero para el resto de valores del potenciómetro, es la función de mapeo la que decide con cuanta intensidad se enciende el LED.

[-] Identification	
Name	AnalogInput
Description	
[-] Data Properties	
Address	400001
Data Type	Word
Client Access	Read/Write
Scan Rate (ms)	2500
[-] Identification	
Name	AnalogOutput
Description	
[-] Data Properties	
Address	400002
Data Type	Word
Client Access	Read/Write
Scan Rate (ms)	2500

Ilustración 218: Mapeo Analógico

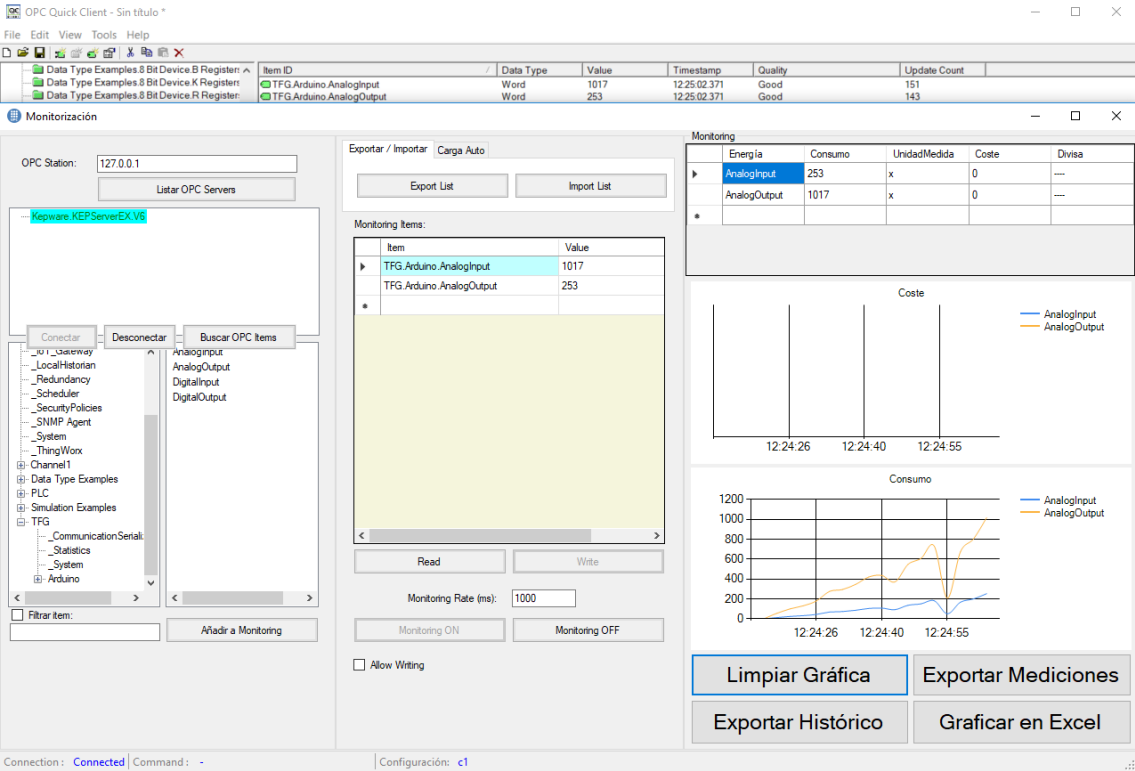


Ilustración 219: Monitorización del Circuito Analógico

Para acabar de testear el circuito analógico, se crean en KEPServer los tags adecuados para los Holding Register, y se elabora una configuración que permita monitorizar en el software desarrollado. Después de poner en marcha la ventana de monitorización, si se juega con el potenciómetro, se aprecia cómo la gráfica varía acorde a la señal que éste suministra al Arduino. Otra vez, es interesante apreciar cómo ambas líneas se dibujan en la gráfica: tienen formas parecidas, pero la pendiente de AnalogOutput siempre es menos pronunciada, además de coincidir en su origen. Se podría concluir, entonces, que el mapeo funciona correctamente.

Reflexiones transversales

Aquí se incluyen las reflexiones sobre la sostenibilidad y el compromiso ambiental, económico y social redactadas para los informes de GEP.

Sostenibilidad y compromiso ambiental

¿Has estimado el impacto ambiental que tendrá la realización del proyecto? ¿Te has planteado minimizar el impacto, por ejemplo, reutilizando recursos?

Este proyecto tiene como objetivo trabajar en un sistema de gestión de recursos a nivel industrial. Actualmente, la industria es una de las mayores fuentes de contaminación del planeta, y uno de los beneficios de sistemas como el que nos ocupa es que tiene como objetivo, ya sea por razones ambientales o económicas, la reducción efectiva del consumo energético. Por tanto, se puede afirmar que el objetivo de este proyecto sí está alineado con la sostenibilidad ambiental, aunque depende del cliente que lo utiliza que el uso sea resulte más o menos provechoso.

En cuanto a la reutilización de recursos, encontramos el gran problema que presenta la industria informática, y es que incluso el equipo más “sencillo” tiene relacionado un proceso de manufactura muy complejo y poco sostenible: los materiales necesarios son derivados del petróleo, emiten sustancias nocivas al tratarlos y los procesos a seguir para transformar las materias primas en piezas funcionales tiende a generar grandes cantidades de residuos.

En las 735h de duración de este proyecto, y teniendo en cuenta que siempre habrá un ordenador encendido consumiendo un promedio de 300W, el consumo eléctrico subiría a 220,5 kWh. Esto supone un gasto inevitable, puesto que es necesario un ordenador para el desarrollo de cada una de las partes de este proyecto.

¿Cómo se resuelven actualmente el problema que quieres abordar (estado del arte)? ¿En qué mejoraría esta solución a las existentes?

La solución propuesta en este proyecto no tiene por qué mejorar ambientalmente a otras alternativas que haya en el mercado, principalmente porque los gastos derivados del desarrollo son similares a la mayoría de proyectos informáticos. Volvemos entonces a la cuestión anterior: depende del cliente donde se despliegue la solución la determinación del beneficio que se obtenga del proyecto.

Todavía no hay alternativas factibles más punteras que la que se propone en éste proyecto. La solución estándar, por ahora, consiste en construir una infraestructura de contadores que lleven las señales a algún tipo de software de gestión, y esto es precisamente lo que se pretende hacer.

Sostenibilidad y compromiso económico

¿Has estimado el coste de la realización del proyecto?

En la sección de presupuesto de esta memoria se especifican los costes de las partes del proyecto, y la inversión en recursos que es necesario hacer. Como se ha hecho con la planificación, se comparará el presupuesto original con el final, y se justificará por qué son diferentes.

¿Cómo se resuelven actualmente el problema que quieres abordar (estado del arte)? ¿En qué mejoraría esta solución a las existentes?

Cómo en el ámbito ambiental, la parte económica de éste proyecto no es diferente a la de otro proyecto similar. En relación a los gastos económicos, se tiene que sumar la energía consumida en el desarrollo del software, así como el coste de adquirir los equipos necesarios para construir la infraestructura, el transporte a las industrias y los salarios del personal. En referencia a esto, el coste no debería de ser un dato notable a tener en cuenta: no hay ningún aspecto que haga prever una inversión desproporcionada de capital, ni se tomará ninguna decisión que haga caer notablemente el coste del proyecto.

En cuanto a la competitividad en el mercado de éste proyecto, se tiene que tener en cuenta que, aunque en gran parte de la solución se utilizan medios de terceros, que se tienen que adquirir, la empresa utiliza recursos propios para otras partes del proyecto, abaratando costes. Es por esto que ésta propuesta pueda tener cierta ventaja respecto de otras alternativas más costosas.

Sostenibilidad y compromiso social

¿Qué crees que te aportará a nivel personal la realización de este proyecto?

Decidí realizar este proyecto en una empresa para poder entrar en contacto con el mundo laboral, e intentar llenar el vacío que se crea entre el aprendizaje en las aulas y el conocimiento adquirido "sobre el terreno". En mi caso, mi primera experiencia laboral: me sirve para poder ver cómo todo aquello que he aprendido en el entorno académico se aplica en la realidad y para consolidar mis conocimientos.

Por otro lado, tener que entrar en una empresa perteneciente a un sector tal al alza como el de la informática industrial hace que necesite aprender a situarme en su contexto corporativo, económico, filosófico y económico. Además, estar rodeado de compañeros de otras nacionalidades y edades es una gran oportunidad para adquirir una perspectiva más amplia de todo, y crecer como persona.

¿Cómo se resuelven actualmente el problema que quieres abordar (estado del arte)? ¿En qué mejoraría esta solución a las existentes?

Teniendo en cuenta el alto nivel de informatización al que están llegando todos los ámbitos de la sociedad, parece inviable que en algunos sectores todavía se usen métodos que no aprovechen de manera exhaustiva el potencial de los computadores. La realidad es que incluso las compañías más grandes y exitosas confían aún en métodos actualmente desfasados. En el caso concreto que nos ocupa, éstos métodos hacen ineficiente el proceso de control de consumos, haciendo trabajar mucho más a los operarios, y dejando los datos desprotegidos ante falsificaciones y otras manipulaciones.

La propuesta que estos sistemas de gestión energética hacen es la de ofrecer un medio a través del cual los operarios puedan monitorizar y gestionar el consumo energético de la industria de manera conveniente e intuitiva, manejando datos en tiempo real, y ofreciendo, todo un abanico de opciones para realizar análisis y reporting con agilidad.

En conclusión, éste proyecto se sitúa en el estado del arte, e intenta satisfacer todas las necesidades que surgen al querer gestionar de manera eficiente el consumo energético de la industria 4.0.

¿Existe una necesidad real del proyecto?

Mientras que el objetivo del proyecto no es intentar llevar el estado del arte más allá, la necesidad industrial que pretende satisfacer corresponde con un problema tan relevante como la sostenibilidad medioambiental.

Algunos de los métodos que se usan en la actualidad para controlar consumos distan mucho de ser perfectos, y comportan notables consecuencias tanto en lo económico como en lo medioambiental y en la sociedad en general. Es por esto que es necesario ofrecer propuestas como este proyecto, ya que constituyen una inversión que produce beneficios una vez amortizada. No solo aumenta los beneficios mediante la corrección de costes, sino que da ejemplo de una manera responsable de trabajar. Además, puede servir de ejemplo para desarrollar soluciones análogas en otros ámbitos, ya sean industriales, comerciales, de investigación o domésticos.

Utilidad de mis estudios universitarios

Indudablemente, mis estudios de grado me han otorgado toda una serie de conocimientos y competencias, que han hecho que pueda llevar a cabo este proyecto, y mi estancia en la empresa, de la manera en la que lo he hecho.

Por un lado, está lo que me han aportado las asignaturas. Por ejemplo, haber estudiado gráficos y diseño de interfaces me ha permitido tanto diseñar el aspecto visual de mi aplicación, como maquetar de manera conveniente la memoria de mi proyecto. Mis conocimientos de redes han hecho posible que pueda entender toda la cadena de comunicaciones que se lleva a cabo en los procesos industriales, desde la captación de señales, hasta la comunicación Ethernet una vez se llega a la red TCP/IP. Por otro lado, las asignaturas de programación me aportaron los conocimientos necesarios para elaborar las estructuras de datos y la jerarquía de clases necesaria, y mis estudios en Sistemas

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

Operativos me dieron el soporte teórico para entender cómo funcionan protocolos como OPC, y para hacer analogías que me permitiesen entender íntegramente cómo funciona la automatización industrial.

De manera más transversal, el grado me ha hecho tener una metodología de trabajo, y entender los problemas y los procesos como un ingeniero debería de hacerlo. Además, el trabajo en competencias transversales me ha dado una amplia sensibilidad en toda una serie de temas, así como en el trabajo en equipo y el autoaprendizaje.

Bibliografía

1. CHEN, J., HE, W., HE, W., & DONG, R. (2010). RESEARCH OF MANUFACTURING EXECUTION SYSTEM SUPPORTING RAPID EXTENDED MANUFACTURING. 2010 2ND INTERNATIONAL CONFERENCE ON INDUSTRIAL AND INFORMATION SYSTEMS. DOI:10.1109/INDUSIS.2010.5565816
2. DAVIES, J. (2013, JUNE 25). INSIDE THE EMERGING ROLE OF CORPORATE ENERGY MANAGERS. RETRIEVED FROM [HTTPS://WWW.GREENBIZ.COM/BLOG/2013/06/25/INSIDE-EMERGING-ROLE-CORPORATE-ENERGY-MANAGERS](https://www.greenbiz.com/blog/2013/06/25/inside-emerging-role-corporate-energy-managers)
3. JUNKER, H., & DOMANN, C. (2017). TOWARDS INDUSTRY 4.0 IN CORPORATE ENERGY MANAGEMENT. ECOSYSTEMS AND SUSTAINABLE DEVELOPMENT XI. DOI:10.2495/ECO170051
4. KEPSERVEREX - OPCFOUNDATION.ORG. (N.D.). RETRIEVED FROM [HTTPS://OPCFoundation.org/products/view/kepserverex](https://opcfoundation.org/products/view/kepserverex)
5. PROJECT SUMMARY. (N.D.). RETRIEVED FROM [HTTP://WWW.IMC-AESOP.ORG/](http://www.imc-aesop.org/)
6. (N.D.). RETRIEVED FROM [HTTP://UMSL.EDU/~SAUTERV/ANALYSIS/6840_F09_PAPERS/NAT/AGILE.HTML](http://umsl.edu/~sauterv/analysis/6840_f09_papers/Nat/Agile.html)
7. AGILE SCRUM METHODOLOGY IN SOFTWARE DEVELOPMENT. (N.D.). RETRIEVED FROM [HTTPS://WWW.SOFTWARETESTINGMATERIAL.COM/AGILE-SCRUM-METHODOLOGY/](https://www.softwaretestingmaterial.com/agile-scrum-methodology/)
8. SIMATIC ENERGY MANAGER – CERTIFIED ENERGY MANAGEMENT ... (N.D.). RETRIEVED FROM [HTTPS://NEW.SIEMENS.COM/GLOBAL/EN/PRODUCTS/AUTOMATION/INDUSTRY-SOFTWARE/AUTOMATION-SOFTWARE/ENERGYMANAGEMENT/SIMATIC-ENERGY-MANAGER-PRO.HTML](https://new.siemens.com/global/en/products/automation/industry-software/automation-software/energymanagement/simatic-energy-manager-pro.html)
9. ALBERTO, J. (2015, AUGUST 07). PERIFERIA E/S CENTRALIZADA VS. PERIFERIA E/S DESCENTRALIZADA CAPÍTULO 2. RETRIEVED FROM [HTTPS://WWW.NOEJU.COM/PERIFERIA-ES-CENTRALIZADA-VS-PERIFERIA-ES-DESCENTRALIZADA-CAPITULO-2/](https://www.noaju.com/periferia-es-centralizada-vs-periferia-es-descentralizada-capitulo-2/)
10. SISTEMA DE CONTROL DISTRIBUIDO. (2019, MARCH 26). RETRIEVED FROM [HTTPS://ES.WIKIPEDIA.ORG/WIKI/SISTEMA_DE_CONTROL_DISTRIBUIDO](https://es.wikipedia.org/wiki/Sistema_de_control_distribuido)
11. ¿QUÉ ES LA PERIFERIA DESCENTRALIZADA? (2018, MAY 10). RETRIEVED FROM [HTTP://WWW.AUTRACEN.COM/LA-PERIFERIA-DESCENTRALIZADA/](http://www.autracen.com/la-periferia-descentralizada/)
12. FLY-BY-WIRE. (2019, JUNE 10). RETRIEVED FROM [HTTPS://EN.WIKIPEDIA.ORG/WIKI/FLY-BY-WIRE](https://en.wikipedia.org/wiki/Fly-by-wire)
13. [HTTPS://CACHE.INDUSTRY.SIEMENS.COM/DL/FILES/442/109742442/ATT_899724/V1/SIMATIC_ENERGY_MANAGER_PRO_OPERATING_MANUAL_EN-US_EN-US.PDF](https://cache.industry.siemens.com/dl/files/442/109742442/att_899724/v1/simatic_energy_manager_pro_operating_manual_en-us_en-us.pdf)
14. 4.0 REASONS WHY EDGE COMPUTING IS RELEVANT FOR INDUSTRY 4.0. (2018, JULY 22). RETRIEVED FROM [HTTPS://IIOT-WORLD.COM/CONNECTED-INDUSTRY/4-0-REASONS-WHY-EDGE-COMPUTING-IS-RELEVANT-FOR-INDUSTRY-4-0/](https://iiot-world.com/connected-industry/4-0-reasons-why-edge-computing-is-relevant-for-industry-4-0/)
15. ADAPTADOR PARA MONTAJE EN PERFIL PAC TMP...7KM9900-0YA00-0AA0. (N.D.). RETRIEVED FROM [HTTPS://RELEPRO.COM/173-ANALIZADOR-DE-RED-7KM-PAC?ORDERBY=PRICE&ORDERWAY=ASC](https://relepro.com/173-analizador-de-red-7km-pac?orderby=price&orderway=asc)
16. BIG DATA. (2019, JUNE 11). RETRIEVED FROM [HTTPS://EN.WIKIPEDIA.ORG/WIKI/BIG_DATA](https://en.wikipedia.org/wiki/Big_data)
17. COMPUTER-INTEGRATED MANUFACTURING. (2019, APRIL 21). RETRIEVED FROM [HTTPS://EN.WIKIPEDIA.ORG/WIKI/COMPUTER-INTEGRATED_MANUFACTURING](https://en.wikipedia.org/wiki/Computer-integrated_manufacturing)

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

18. CONSEQUENCES OF INDUSTRY 4.0 IN BUSINESS AND ECONOMICS. (N.D.). RETRIEVED FROM
[HTTPS://WWW.RESEARCHGATE.NET/PUBLICATION/326943099_CONSEQUENCES_OF_INDUSTRY_40_IN_BUSINESS_AND_ECONOMICS](https://www.researchgate.net/publication/326943099_CONSEQUENCES_OF_INDUSTRY_40_IN_BUSINESS_AND_ECONOMICS)
19. DESIGN ZONE - DESIGN GUIDES. (2019, APRIL 05). RETRIEVED FROM
[HTTPS://WWW.CISCO.COM/C/EN/US/SOLUTIONS/DESIGN-ZONE.HTML](https://www.cisco.com/c/en/us/solutions/design-zone.html)
20. DIGITAL MODELING AND FABRICATION. (2019, APRIL 25). RETRIEVED FROM
[HTTPS://EN.WIKIPEDIA.ORG/WIKI/DIGITAL_MODELING_AND_FABRICATION](https://en.wikipedia.org/wiki/Digital_modeling_and_fabrication)
21. HIGH-AVAILABILITY SEAMLESS REDUNDANCY. (2019, MAY 08). RETRIEVED FROM
[HTTPS://EN.WIKIPEDIA.ORG/WIKI/HIGH-AVAILABILITY_SEAMLESS_REDUNDANCY](https://en.wikipedia.org/wiki/High-availability_seamless_redundancy)
22. HOW WILL AUTOMATION AFFECT ECONOMIES AROUND THE WORLD? (N.D.). RETRIEVED FROM
[HTTPS://WWW.MCKINSEY.COM/FEATURED-INSIGHTS/FUTURE-OF-WORK/HOW-WILL-AUTOMATION-AFFECT-ECONOMIES-AROUND-THE-WORLD](https://www.mckinsey.com/featured-insights/future-of-work/how-will-automation-affect-economies-around-the-world)
23. INDUSTRIAL INTERNET OF THINGS. (2019, JUNE 04). RETRIEVED FROM
[HTTPS://EN.WIKIPEDIA.ORG/WIKI/INDUSTRIAL_INTERNET_OF_THINGS](https://en.wikipedia.org/wiki/Industrial_Internet_of_Things)
24. INDUSTRIE 4.0 AND THE FACTORY NETWORK. (N.D.). RETRIEVED FROM
[HTTP://I40TODAY.COM/CISCO/](http://i40today.com/cisco/)
25. INDUSTRY 4.0 - THE NINE TECHNOLOGIES TRANSFORMING INDUSTRIAL PRODUCTION. (N.D.). RETRIEVED FROM
[HTTPS://WWW.BCG.COM/CAPABILITIES/OPERATIONS/EMBRACING-INDUSTRY-4.0-REDISCOVERING-GROWTH.ASPX](https://www.bcg.com/capabilities/operations/embracing-industry-4.0-rediscovering-growth.aspx)
26. INFORMACIÓN DETALLADA SOBRE EL PROTOCOLO MODBUS. (N.D.). RETRIEVED FROM
[HTTP://WWW.NI.COM/ES-ES/INNOVATIONS/WHITE-PAPERS/14/THE-MODBUS-PROTOCOL-IN-DEPTH.HTML](http://www.ni.com/es-es/innovations/white-papers/14/the-modbus-protocol-in-depth.html)
27. ISO 50001. (N.D.). RETRIEVED FROM
[HTTP://OCACERT.COM/CERTIFICACIONISO50001.PHP?GLID=EAIaIQobChMItaCBQDPm4GIVigzTCH1mHGWdEAAYAAAEGLCPD_BwE](http://ocacert.com/certificacionISO50001.php?gclid=EAIaIQobChMItaCBQDPm4GIVigzTCH1mHGWdEAAYAAAEGLCPD_BwE)
28. ISO/IEC 21827:2008. (2014, AUGUST 28). RETRIEVED FROM
[HTTPS://WWW.ISO.ORG/STANDARD/44716.HTML](https://www.iso.org/standard/44716.html)
29. LABS, W., & LABS, W. (2017, NOVEMBER 14). 'INDUSTRY 4.0' NETWORK ARCHITECTURE RELIES ON INTERCONNECTIVITY. RETRIEVED FROM
[HTTPS://WWW.FOODENGINEERINGMAG.COM/ARTICLES/97066-INDUSTRY-40-NETWORK-ARCHITECTURE-RELIES-ON-INTERCONNECTIVITY](https://www.foodengineeringmag.com/articles/97066-industry-40-network-architecture-relies-on-interconnectivity)
30. LEARN- & TRAINING DOCUMENTS - SIEMENS AUTOMATION COOPERATES WITH EDUCATION. (N.D.). RETRIEVED FROM
[HTTPS://W3.SIEMENS.COM/MCMS/SCE/EN/ADVANCED_TRAINING/TRAINING_MATERIAL/PAGES/DEFAULT.ASPX](https://w3.siemens.com/mcms/sce/en/advanced_training/training_material/pages/default.aspx)
31. MACHINE TO MACHINE. (2019, JUNE 10). RETRIEVED FROM
[HTTPS://EN.WIKIPEDIA.ORG/WIKI/MACHINE_TO_MACHINE](https://en.wikipedia.org/wiki/Machine_to_machine)
32. MODBUS. (2019, MARCH 04). RETRIEVED FROM
[HTTPS://ES.WIKIPEDIA.ORG/WIKI/MODBUS](https://es.wikipedia.org/wiki/Modbus)
33. NETWORKING AND SECURITY IN INDUSTRIAL AUTOMATION ENVIRONMENTS DESIGN AND IMPLEMENTATION GUIDE - NETWORKING AND SECURITY IN INDUSTRIAL AUTOMATION ENVIRONMENTS [INDUSTRIAL NETWORKING]. (2019, JUNE 07). RETRIEVED FROM
[HTTPS://WWW.CISCO.COM/C/EN/US/TD/DOCS/SOLUTIONS/VERTICALS/INDUSTRIAL_AU](https://www.cisco.com/c/en/us/td/docs/solutions/verticals/industrial_aut)

ESTUDIO I PROGRAMACIÓN DE UN SISTEMA MES QUE MONITORIZA I CONTROLA EL CONSUMO DE ENERGÍA DE PROCESOS INDUSTRIALES

- TOMATION/IA_HORIZONTAL/DG/INDUSTRIAL-AUTOMATIONDG/INDUSTRIAL-AUTOMATIONDG.HTML
34. OTHER RESOURCES INDUSTRY 4.0 AND THE CONSEQUENCES FOR THE LABOR MARKET AND ECONOMY. (N.D.). RETRIEVED FROM [HTTPS://SKILLSPANORAMA.CEDEFOP.EUROPA.EU/EN/USEFUL_RESOURCES/INDUSTRY-40-AND-CONSEQUENCES-LABOR-MARKET-AND-ECONOMY](https://skillspanorama.cedefop.europa.eu/en/useful_resources/industry-40-and-consequences-labor-market-and-economy)
 35. PARALLEL REDUNDANCY PROTOCOL. (2018, MARCH 12). RETRIEVED FROM [HTTPS://EN.WIKIPEDIA.ORG/WIKI/PARALLEL_REDUNDANCY_PROTOCOL](https://en.wikipedia.org/wiki/Parallel_Redundancy_Protocol)
 36. PRIYANKARA, N. P. (N.D.). WRITE TO A FILE USING CSVHELPER IN C#. RETRIEVED FROM [HTTPS://STACKOVERFLOW.COM/QUESTIONS/23192696/WRITE-TO-A-FILE-USING-CSVHELPER-IN-C-SHARP](https://stackoverflow.com/questions/23192696/write-to-a-file-using-csvhelper-in-c-sharp)
 37. RATHEE, P. R. (N.D.). JSON STRING TO CSV AND CSV TO JSON CONVERSION IN C#. RETRIEVED FROM [HTTPS://STACKOVERFLOW.COM/QUESTIONS/36274948/JSON-STRING-TO-CSV-AND-CSV-TO-JSON-CONVERSION-IN-C-SHARP](https://stackoverflow.com/questions/36274948/json-string-to-csv-and-csv-to-json-conversion-in-c-sharp)
 38. RTUS FOR INDUSTRY. (N.D.). RETRIEVED FROM [HTTPS://W3.SIEMENS.COM/MCMS/PROCESS-CONTROL-SYSTEMS/SITECOLLECTIONDOCUMENTS/EFILES/PCS7/PDF/76/PRDBRIEF/BR_RTU_EN.PDF](https://w3.siemens.com/mcms/process-control-systems/sitecollectiondocuments/efiles/pcs7/pdf/76/prdbrief/br_rtu_en.pdf)
 39. SEGUIR. (N.D.). INDUSTRY 4 NETWORK DESIGN AND SECURITY - PETER BROWN. RETRIEVED FROM [HTTPS://ES.SLIDESHARE.NET/PROFIBUSUK/INDUSTRY-4-NETWORK-DESIGN-AND-SECURITY-PETER-BROWN](https://es.slideshare.net/ProfibusUK/industry-4-network-design-and-security-peter-brown)
 40. SIMATIC ENERGY MANAGER PRO OPERATION. (N.D.). RETRIEVED FROM [HTTPS://CACHE.INDUSTRY.SIEMENS.COM/DL/FILES/442/109742442/ATT_899724/V1/SIMATIC_ENERGY_MANAGER_PRO_OPERATING_MANUAL_EN-US_EN-US.PDF](https://cache.industry.siemens.com/dl/files/442/109742442/att_899724/v1/simatic_energy_manager_pro_operating_manual_en-us_en-us.pdf)
 41. THE IMC-AESOP ARCHITECTURE FOR CLOUD-BASED INDUSTRIAL CPS. (N.D.). RETRIEVED FROM [HTTPS://WWW.RESEARCHGATE.NET/PUBLICATION/271764201_THE_IMC-AESOP_ARCHITECTURE_FOR_CLOUD-BASED_INDUSTRIAL_CPS](https://www.researchgate.net/publication/271764201_the_imc-aesop_architecture_for_cloud-based_industrial_cps)
 42. TIME-SENSITIVE NETWORKING. (2019, MAY 18). RETRIEVED FROM [HTTPS://EN.WIKIPEDIA.ORG/WIKI/TIME-SENSITIVE_NETWORKING](https://en.wikipedia.org/wiki/Time-sensitive_networking)
 43. VIAR, & VIAR. (2018, APRIL 22). A FEW REAL-WORLD EXAMPLES OF INDUSTRY 4.0. RETRIEVED FROM [HTTPS://MEDIUM.COM/@VIARBOX/A-FEW-REAL-WORLD-EXAMPLES-OF-INDUSTRY-4-0-8E2DE4F4F23E](https://medium.com/@viarbox/a-few-real-world-examples-of-industry-4-0-8e2de4f4f23e)
 44. WHAT IS BIG DATA ANALYTICS? - DEFINITION FROM WHATIS.COM. (N.D.). RETRIEVED FROM [HTTPS://SEARCHBUSINESSANALYTICS.TECHTARGET.COM/DEFINITION/BIG-DATA-ANALYTICS](https://searchbusinessanalytics.techtarget.com/definition/big-data-analytics)
 45. WONG, C. (2019, MARCH 29). INDUSTRY 4.0 COULD CREATE MILLIONS OF NEW JOBS. RETRIEVED FROM [HTTPS://WWW.FUTURITHMIC.COM/2019/02/13/INDUSTRY-4-0-COULD-CREATE-MILLIONS-NEW-JOBS/](https://www.futurithmic.com/2019/02/13/industry-4-0-could-create-millions-new-jobs/)